An Investigation of Cardan Rotation Sequences on Lumbar Spine Lifting Kinematics

by
Ednah Grace Louie
B.A., Mount Holyoke College, 2014

Submitted to the graduate degree program of Bioengineering and the Graduate Facility of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____
Committee Chair: Sara Wilson

_____
Committee Member: Trent Herda

_____
Committee Member: Carl Luchies

Date Defended: 31 August 2018

The Thesis Committee for Ednah Grace Louie

certifies that this is the approved version of the following thesis:

An Investigation of Cardan Rotation Sequences on Lumbar Spine Lifting Kinematics

_____

Committee Chair: Sara Wilson

_____

Committee Member: Trent Herda

_____

Committee Member: Carl Luchies

Date Approved:

31 August 2018

ABSTRACT

Euler angles are commonly used for kinematic descriptions due to their ability to translate to clinical contexts and the conciseness of using three rotation angles. A challenge with using Euler angles is their dependency on Euler sequence selection. Many studies do not use the same Euler sequence, leading to difficulty comparing across studies. Though sequence selection recommendations have been made, there are disagreements among researchers on these recommendations. The goal of this study is to assess the effect of Cardan sequence selection on lumbar spine motions restricted to a single plane and motions across multiple planes. Cardan sequences are a subset of Euler sequences using all three orthogonal directions. This goal was investigated through collecting the lumbar spine kinematics of 22 human participants performing cyclic lifting tasks in two positions and three directions. Four spine angles were calculated: trunk flexion, lumbar flexion, lumbar lateral rotation, and lumbar axial rotation. It was hypothesized that motion restricted to a single plane would be best represented by sequences where the first rotation matches the plane of motion. Through two lifting tasks restricted to the sagittal plane, this study found sequences other than YXZ and ZXY could describe the four calculated spine angles. It was also hypothesized that motions occurring across multiple planes would be best represented by the sequence with the planes of motion ordered by the magnitude of range of motion, from largest to smallest. This is supported by investigation of four asymmetric lifting tasks where sequences XYZ and XZY were found to give the best representation of the motions performed. In these asymmetric lifting tasks, motion about X was the greatest, and the magnitudes of motion about Y and Z were very similar. When assessing the robustness of Cardan sequences for description of lumbar spine angles across single and multiple-plane lifting motions, we would recommend XYZ or XZY as the sequence of choice.

ACKNOWLEDGEMENTS

To God: who brought me on this graduate journey with more twists than expected. Through Him all things begun are finished.

To Dr. Sara Wilson: for her support and grace working with me through the ups, downs, and changes in my graduate journey and research.

To the Madison and Lila Self Graduate Fellowship: for supporting three years of my graduate research assistantship.

To S. Mukui Mutunga: for her friendship and support as a researcher throughout the whole thesis. She was essential during data collection and a valuable sounding board for my thoughts.

To Anthony Caruso and Jessica Kirchner: for their assistance during data collection as volunteer undergraduate research assistants.

Finally, there are more people than I can succinctly list who have assisted through belief in me, emotional support, motivation, and accountability. Among them, I would like to specifically acknowledge Mukui, Bhargavi Krishnan, Jenelle Hallaert, Karen Prescott, Edward Louie, David and Teresa Louie, Rachel Roca, and Samuel Davis: for their contributions of realistic perspectives and practical support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1: Motivation

The prevalence and productivity costs of low back pain have motivated many low back and spine motion studies over the years. Over 80% of adults will experience low back pain (LBP) at some point in their lives and account for billions of dollars in lost productivity each year[5,9]. Occupational factors associated with increased risk of LBP include exposure to heavy loads, prolonged periods of trunk flexion, repetitive motions involving twisting of the torso, lifting, and whole-body vibrations.

While lumbar spine research has increased, differences in kinematics measurement and calculation methods have presented challenges in repeatability and comparisons across studies. In recent years, the National Institutes of Health (NIH) has highlighted its interest in LBP, and noted the challenges researchers faced finding strategies for LBP prevention and treatment which included difficulties comparing results due to inconsistent terminology and outcome measures[12]. This paper will focus on comparing methods to assess 3D kinematics of spine motion during repetitive lifting tasks in different directions and initial positions.

1.2: Background

1.2.a: Spine Motions

Spine movement occurs as combinations across three anatomical planes: sagittal, frontal, and transverse (Fig. 1.1). The spine can be divided into four sections based on its curvature: cervical, thoracic, lumbar, and sacral spine (Fig. 1.2). Spinal rotations occurring in the sagittal plane can be called flexion/extension. Lumbar flexion/extension can range from kyphotic to lordotic with a neutral posture being slightly lordotic (Fig. 1.3). Lumbar rotation in the frontal

plane can be called left/right lateral bending, and rotation in the transverse plane is called left/right axial rotation (Fig. 1.4).

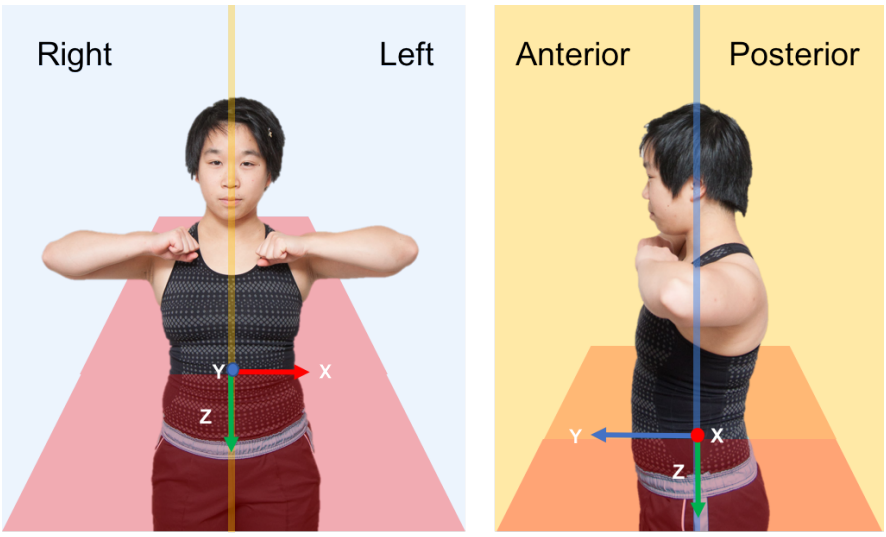

Fig. 1.1: Planes of rotation: sagittal (yellow), frontal (blue), and transverse (red). The local axes used for this paper are shown, following the right-hand rule: x-axis (red, medially right to left), y-axis (blue, posterior to anterior), and z-axis (green, superior to inferior).

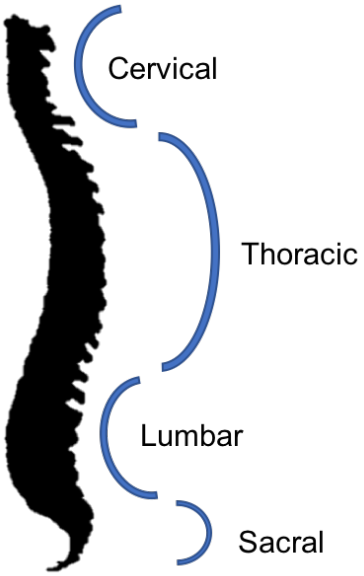

Fig. 1.2: Sections of the spine and their curvature

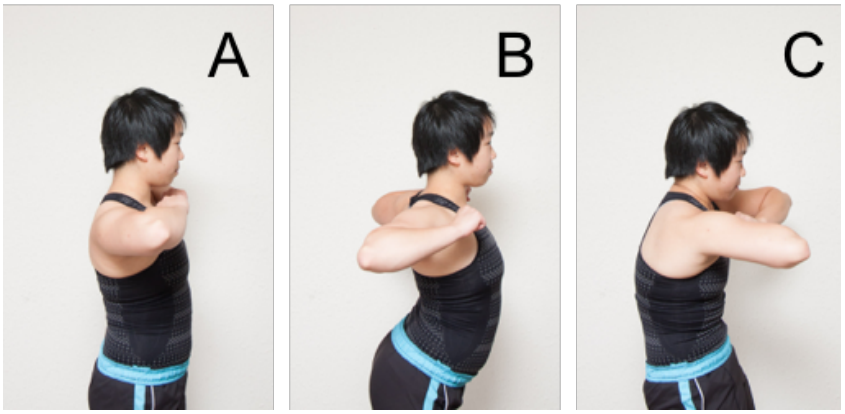

Fig. 1.3: Lumbar rotation in the sagittal plane describing lumbar posture: neutral upright, slightly lordotic (A), more lordotic (B), and more kyphotic (C)

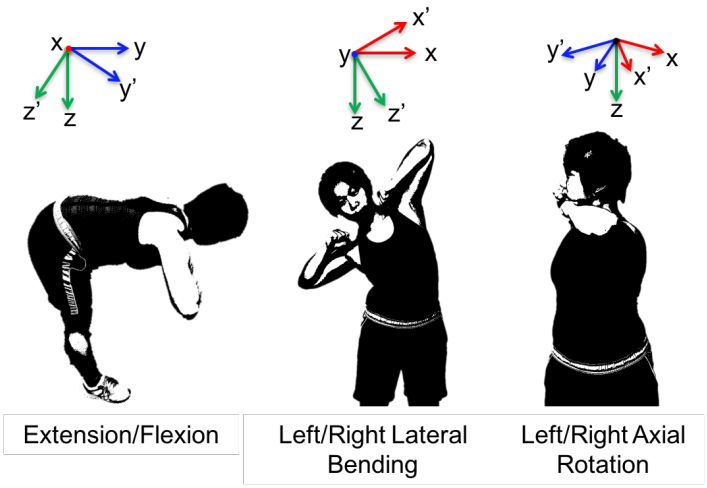

Fig. 1.4: Spine rotation in the sagittal plane (extension/flexion), frontal plane (left/right lateral bending), and transverse plane (left/right axial rotation)

1.2.b: Euler Angles and Euler/Cardan Sequences

Currently, there are a number of methods available for researchers investigating the mechanics of the spine and pelvis. Three-dimensional joint motion can be described in a variety of ways. Euler angles, helical axes, and projection planes are just a few common methods. The choice of kinematic analysis method varies among research studies and groups even as establishing

standards are discussed. However, the use of Euler angles remains the most common method in biomechanics due to its minimal numerical description and factors affecting other methods such as noise.

Euler angles are a system of three angles describing the change in an object's orientation relative to a fixed coordinate system. These three angles are derived from a Euler sequence of three rotations about axes of the fixed coordinate system to express the change from an initial ***x***, ***y***, ***z*** orientation to a final ***x'***, ***y'***, ***z'*** orientation. For example, a YXZ sequence will describe an orientation change by first rotating the local coordinate axis about its y-axis by R_y, followed by a rotation R_x about the re-oriented x-axis, and ending with a rotation R_z about the final z-axis. These angles are commonly denoted as roll, pitch, yaw. In human motion studies, these angles are often defined as flexion/extension in the sagittal plane, lateral rotation in the frontal plane, and axial rotation in the transverse plane (Fig. 1.4).

Euler angles are extracted from elements of the 3x3 matrix resulting from the ordered multiplication of the three rotation matrices. Rotation about each axis are modelled by rotation matrices,

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplication of the three matrices in an ordered sequence will produce a matrix such as

$$R_{xyz} = \begin{bmatrix} \cos\beta\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\beta\sin\gamma & \cos\alpha\sin\beta \\ \cos\alpha\sin\gamma & \cos\alpha\cos\gamma & -\sin\alpha \\ \sin\alpha\cos\beta\sin\gamma - \sin\beta\cos\gamma & \sin\alpha\cos\beta\cos\gamma + \sin\beta\sin\gamma & \cos\alpha\cos\beta \end{bmatrix}$$

from which the values for Euler angles α, β, and γ may be extracted:

$$\alpha = -\sin^{-1}(R_{xyz}(2,3))$$

$$\frac{(R_{xyz}(1,3))}{(R_{xyz}(3,3))} = \frac{\cos\alpha \sin\beta}{\cos\alpha \cos\beta} = \tan\beta \therefore \beta = \tan^{-1}\frac{(R_{xyz}(1,3))}{(R_{xyz}(3,3))}$$

$$\frac{(R_{xyz}(2,1))}{(R_{xyz}(2,2))} = \frac{\cos\alpha \sin\gamma}{\cos\alpha \cos\gamma} = \tan\gamma \therefore \gamma = \tan^{-1}\frac{(R_{xyz}(2,1))}{(R_{xyz}(2,2))}$$

Cardan sequences are a subset of Euler sequences: while Euler angles may result from any combination of the three axes (e.g., XYX, YZX, or YZY), Cardan angles result from strict combinations of rotations about three *orthogonal* axes (e.g., ZYX or YXZ, and not XZX). As such, there are six possible Cardan sequences: XYZ, XZY, YXZ, YZX, ZXY, and ZYX. The study in this paper will focus on these six Cardan sequences.

Differences in Euler/Cardan sequence have been shown to result in significantly different values of angular motion[1,3]. Attempts to standardize Euler/Cardan sequences for different joints have been made, but the adoption of these standards has varied. Different definitions of coordinate axes also exist, creating further confusion and inconsistency among recommendations and studies (Fig. 1.5). It is thus essential for Euler/Cardan sequence(s) and coordinate axes definitions to always be reported in kinematic studies using this method.

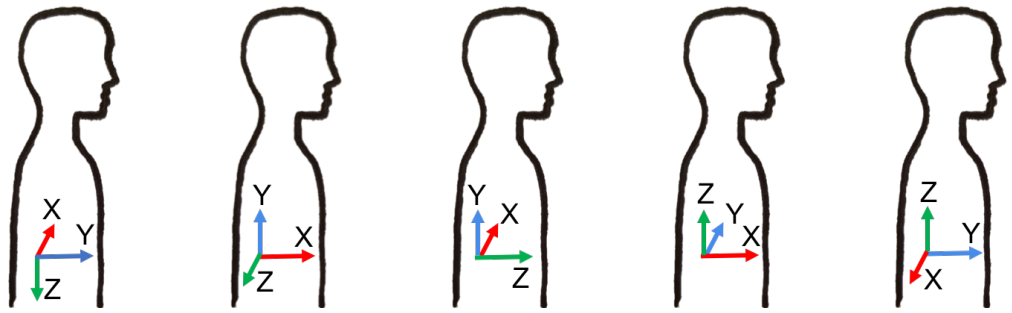

Fig. 1.5: A few of the different local coordinate axes definitions used and/or recommended by different organizations and research groups.

For the rest of this paper, the positive coordinate axes shall be defined as: the x-axis from right to left, the y-axis from posterior to anterior, and the z-axis superior to inferior (Fig. 1.1). As such, rotation about the x-axis would describe flexion/extension (motion in the sagittal plane), lateral bending about the y-axis (motion in the frontal plane), and axial twist about the z-axis (motion in the transverse plane). This axis definition is consistent with axis definitions used in previous studies conducted in this laboratory, and matches the definition set in the Motion Monitor (Innsport, IL) system used in this laboratory. The Cardan sequence considered in previous studies will be, to the closest comparison, translated to match the axes defined above. A sagittal-frontal-transverse sequence would thus be translated as XYZ.

1.2.c: Gimbal Lock

Apart from the effect of sequence selection, another flaw that occurs with using Euler angles is the occurrence of a gimbal lock. The effect of gimbal lock leads to a loss of an axis description when the axis crosses parallel with another axis. Most often gimbal lock occurs when

the middle rotation approaches ± 90°[13]. By description of the output Euler angles in kinematics studies, this looks like an instantaneous jump of a multiple of ± 90° occurs: common values of gimbal lock jumps include ± 90°, ± 180°, and ± 360°.

Steps to minimize or correct for gimbal lock occurrences exist. Output of gimbal lock can be corrected by post-data collection analysis code. Corrections tend to look for instantaneous changes in a range of magnitude, and apply a correction for the appropriate multiple of ± 90°. Instances of gimbal lock may be minimized by restricting movement to ranges that do not put the middle rotation near 90°[13].

1.2.d: Previous Studies

As the challenges of differing definitions and methodologies became an apparent issue, studies recommending standardization were made. In 1995 the International Society of Biomechanics (ISB) provided their initial recommendations for standardizing reports of kinematic data based on Grood and Suntay's Joint Coordinate System (JCS)[2]. The JCS for a joint is created by defining an origin and two axes based on anatomical landmarks, with a third, orthogonal, axis resulting from the cross product of the two axes. The third axis is called the floating axis as it is not fixed and moves relative to the two defined axes. The ISB formed subcommittees of volunteer researchers specializing in specific joints to determine the recommendations for each joint. In their 1995 report, the ISB recommended an XZY sequence to describe the orientation transformation from the local coordinate axis to the global reference frame[2]. Technical reports published in 2002 and 2005 provided recommendations for defining joint coordinate systems on various joints for studies on human motion based on anatomical landmarks[11,14]. Recommendations for the spine defined the local axis based on vertebral landmarks: the y-axis is formed by a line "passing through

the centers of the vertebra's upper and lower endplates", and the z-axis formed from the locations of "the bases of the right and left pedicles"[11]. These definitions are often used in orthopedic and cadaveric research, where locating such bony landmarks is accessible through MRI or CT imaging, or on exposed vertebrae. This definition is not practical for many human motion studies: many research labs use motion-tracking sensors attached to the skin surface and access to the bony landmarks or rigid attachment to the vertebrae would be invasive. Additionally, some labs lack access to equipment to conduct motion studies within a CT or MRI modality, or the motions studied are not appropriate to be performed within the modality.

Despite recommendations on Euler/Cardan sequence selection (Table 1), the reality is there are still disagreements in practice due to different defining criteria when assessing sequences, differences in defining coordinate axes, and different motions assessed. The ISB's 1995 report noted a number of contemporary standardization attempts being pursued by organizations like the Scoliosis Research Society, the Clinical Gait Laboratory Group, and through the European Computer Aided Movement Analysis in a Rehabilitation Context (CAMARC) Project; their 2002 report noted at least two other axis conventions commonly used, and noted a need for further tests of their recommendations for verification and revision[11]. In a 1993 study, Nowinski et al recommended sequence selection be made with consideration of the magnitude of motion taken in each plane[15]. For example, if a motion is predominantly in the sagittal plane, then the frontal plane, and finally the transverse plane, XYZ may be the most appropriate choice. A 1996 study by Crawford et al on the six Cardan sequences and projection angles using data collected by Panjabi et al[6] recommended ZYX for describing spine motion through considerations of vertebral symmetry[1]. Baker conducted his 2001 computational study with the criteria that the recommended sequence be able to correspond to clinical definitions and understanding of angles and

recommended a ZXY sequence[7]. For the cervical spine, Skalli et al recommended the YXZ sequence[3] following a computational study with a vertebral model. Still others recommend XZY for the cervical spine[16-18].

Table 1: Summary of different recommended Cardan sequences for spine kinematics based on axes definitions in Fig. 1

ISB[2]	Crawford[1], Panjabi	Baker[7]	Skalli[3]	Ishii[16], Lin[17], Sugiura[18]	Lees[4]
XZY	ZYX	ZXY, or XYZ	YXZ	XZY	XYZ or XZY

The impact of Euler/Cardan angle selection is also affected by the specific joint and motion being studied. Anderst et al found inherent anatomical and kinematic variability between subjects' mean differences due to calculation for the group average spinal ROM was only significant in small sample sizes for cervical spine motion during dynamic head and neck ROM movements[10]. Baker's study noted the commonly used XYZ sequence was similarly robust to his recommended ZXY if values for pelvic torsion and anterior-posterior tilt were small[7]. Studies on Cardan angles in lumbar-pelvic kinematics during gait and running found deviations due to sequence selection were insignificant, as these activities often saw spinal rotations less than 10°[3,8] and are primarily limited to motion in a single plane[1]. However, motions where the spine rotates 30° or more show significant divergence due to Cardan angle sequence[1,3].

The effect of Euler/Cardan sequence selection has been examined for a number of joints and motions, but few studies of any joint consider all six Cardan sequences, and even fewer for more complex, dynamic motions. Some studies only compare two or three Euler/Cardan sequences, or compare a particular sequence to alternative calculation methods. For his computational study holding different pelvic angles constant, Baker assessed the ZXY and XYZ

sequences[7]. A dynamic motion study by Lees et al compared the effect of the six Cardan sequences on orientation angles in the ankle, knee, and hip of the support leg during a maximal instep soccer kick. When different sequences were applied in analysis, the resulting dynamic of Euler angles describing motions outside of the sagittal plane (abduction/addiction and internal/external rotations) varied in shape and offset magnitude[4]. The number of comprehensive studies is still sparse, and Anderst et al's 2017 study noted a "lack of in vivo data to compare calculation methods"[10].

The few studies on Euler/Cardan sequence selection on spine motions have alluded to significant divergences for large spine motions with rotations of 30° or more[1,3]. It is thus important to systematically assess the effect of Euler/Cardan angle sequence selection on large and complex spine motions crossing moving dynamically through multiple planes as well as its effect on simple spine motions occurring mostly within one plane.

1.3: Specific Aims

The overall goal of this work is to investigate the effect of Cardan rotation sequence on the lumbar spine motion restricted to one plane and across multiple planes. We hypothesized that (1) motion restricted to a single plane would be best represented by Cardan rotation sequences where the first rotation matches the plane of motion; (2) motion occurring across multiple planes would be best represented by the Cardan rotation sequence with the planes of motion ordered by magnitude of range of motion, from largest to smallest.

To investigate the effect of Euler rotation sequence choice on spine motion, a study of 23 human subjects performing six different lifting tasks was conducted to determine the effect of Cardan rotation sequences on symmetric and asymmetric (coupled) motions in the lumbar spine.

1.4: Thesis Content

This document consists of three chapters. Chapter 1 establishes the context of the research studies through a summary of relevant published literature and outlines the subsequent content. Chapter 2 consists of a proposed journal article investigating the effect of Cardan sequence choice on lifting motions in the lumbar spine. Finally, Chapter 3 summarizes the body of this work.

1.5: References

1. Crawford NR, Yamaguchi, G.T., Dickman, C.A. Methods for determining spinal flexion/extension, lateral bending, and axial rotation from marker coordinate data: Analysis and refinement. Human Movement Science 1996;15:55-78.
2. Wu G, Cavanagh PR. ISB recommendations for standardization in the reporting of kinematic data. J Biomech 1995;28(10):1257-61.
3. Skalli W, Lavaste F, Descrimes JL. Quantification of three-dimensional vertebral rotations in scoliosis: what are the true values? Spine (Phila Pa 1976) 1995;20(5):546-53.
4. Lees A, Barton G, Robinson M. The influence of Cardan rotation sequence on angular orientation data for the lower limb in the soccer kick. J Sports Sci 2010;28(4):445-50.
5. Deyo RA, Weinstein JN. Low back pain. N Engl J Med 2001;344(5):363-70.
6. Panjabi MM, Oda T, Crisco JJ, 3rd, Dvorak J, Grob D. Posture affects motion coupling patterns of the upper cervical spine. J Orthop Res 1993;11(4):525-36.
7. Baker R. Pelvic angles: a mathematically rigorous definition which is consistent with a conventional clinical understanding of the terms. Gait Posture 2001;13(1):1-6.
8. Schache AG, Wrigley TV, Blanch PD, Starr R, Rath DA, Bennell KL. The effect of differing Cardan angle sequences on three dimensional lumbo-pelvic angular kinematics during running. Med Eng Phys 2001;23(7):493-501.
9. Maetzel A, Li L. The economic burden of low back pain: a review of studies published between 1996 and 2001. Best Pract Res Clin Rheumatol 2002;16(1):23-30.
10. Anderst WJ, Aucie Y. Three-dimensional intervertebral range of motion in the cervical spine: Does the method of calculation matter? Med Eng Phys 2017;41:109-115.
11. Wu G, Siegler S, Allard P, Kirtley C, Leardini A, Rosenbaum D, Whittle M, D'Lima DD, Cristofolini L, Witte H and others. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion--part I: ankle, hip, and spine. International Society of Biomechanics. J Biomech 2002;35(4):543-8.
12. Deyo RA, Dworkin SF, Amtmann D, Andersson G, Borenstein D, Carragee E, Carrino J, Chou R, Cook K, DeLitto A and others. Report of the Task Force on Research Standards for Chronic Low-Back Pain; 2013.
13. Vass G. Avoiding Gimbal Lock. Computer Graphics World 2009:10-11.
14. Wu G, van der Helm FC, Veeger HE, Makhsous M, Van Roy P, Anglin C, Nagels J, Karduna AR, McQuade K, Wang X and others. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion--Part II: shoulder, elbow, wrist and hand. J Biomech 2005;38(5):981-992.
15. Nowinski GP, Visarius H, Nolte LP, Herkowitz HN. A biomechanical comparison of cervical laminaplasty and cervical laminectomy with progressive facetectomy. Spine (Phila Pa 1976) 1993;18(14):1995-2004.
16. Ishii T, Mukai Y, Hosono N, Sakaura H, Fujii R, Nakajima Y, Tamura S, Sugamoto K, Yoshikawa H. Kinematics of the subaxial cervical spine in rotation in vivo three-dimensional analysis. Spine (Phila Pa 1976) 2004;29(24):2826-31.
17. Lin CC, Lu TW, Wang TM, Hsu CY, Hsu SJ, Shih TF. In vivo three-dimensional intervertebral kinematics of the subaxial cervical spine during seated axial rotation and lateral bending via a fluoroscopy-to-CT registration approach. J Biomech 2014;47(13):3310-7.

18. Sugiura T, Nagamoto Y, Iwasaki M, Kashii M, Kaito T, Murase T, Tomita T, Yoshikawa H, Sugamoto K. In vivo 3D kinematics of the upper cervical spine during head rotation in rheumatoid arthritis. J Neurosurg Spine 2014;20(4):404-10.

Chapter 2: The Effect of Cardan Sequence on Descriptions of Symmetric and Asymmetric Lumbar Lifting Motions

2.1: Introduction

The range of lumbar spine motion has been well-quantified and characterized over many years of studies[1-5]. Coordination of lumbar posture during activities that load the spine is of great interest due to high prevalence of low back pain in the populace. While there are strong motivations for studying spine motion, differences in kinematic calculation methodology pose a challenge in comparing results across studies and research groups. In response, the International Society of Biomechanics (ISB) proposed a standard for reporting kinematic data in 1995[6]. Contemporaneous with the ISB, other groups across the world also pursued studies to recommend methodology standards.

Euler angles are commonly used in kinematic calculations due to their ability to translate to clinical contexts and the concise use of three position values and three angles describing orientation. A challenge with this method is dependency on sequence selection. Euler sequences include permutations of rotations about the x-, y-, and z-axis (e.g. XYZ, YZX, etc) and rotations about two axes where rotation about one axis is taken twice (e.g. XYX, ZYZ, etc). Euler sequences using all three orthogonal axes (e.g. YXZ) form a subset of six sequences called Cardan sequences. In the process of determining methodology standards, researchers have examined the effect of Euler/Cardan sequences on motions of various joints and limbs, ranging from the shoulder and spine to the pelvis, knee, and ankle[7-10]. These studies have shed greater understanding on the benefits and problems of different sequences and calculation methods. However, standardization has yet to settle conclusively due to disagreement in definitions and recommendations, as well as applicability to different joints and motions in human subject, cadaveric, or computational studies. For the rest of this paper, unless otherwise noted, all described sequences, rotations, and axis

directions will be based on the ones shown in Figure 2.1. The axes are defined as: x (right to left), y (posterior to anterior, and z (superior to inferior).

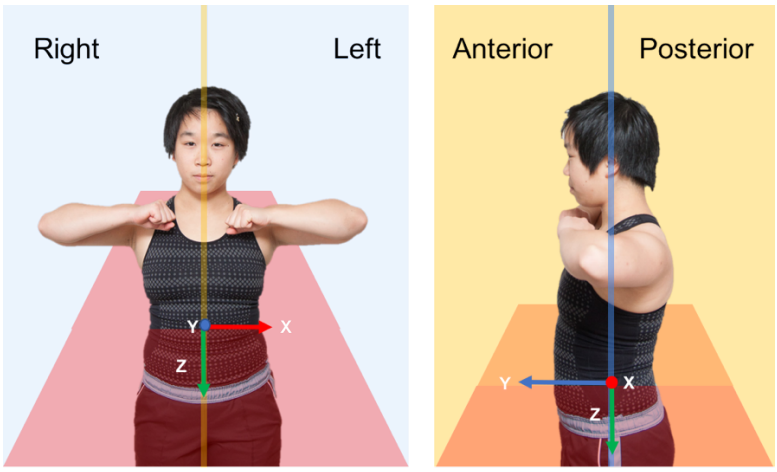

Fig. 2.1: Planes and axes definitions for this paper. For planes: sagittal (yellow), frontal (blue), and transverse (red). For axes: x-axis (red, right to left), y-axis (blue, posterior to anterior), and z-axis (green, superior to inferior).

Studies examining the effect of Euler/Cardan sequences on spine motions have drawn a number of conclusions, but some disagreements and gaps still prevail. It is generally agreed that choice in sequence does not greatly change the value of Euler angles for tasks that do not require large changes in spine motion. In a treadmill running study with four subjects, Schache et al found Cardan sequences did not have effects exceeding 7.0° and 28° for the lumbar spine and pelvic rotation, respectively[11]. Baker corroborated this observation in a 2001 computational analysis comparing two different sequences on the pelvis[7]. His case study indicated that the ZXY sequence is robust and potentially improved over the commonly-used XYZ sequence for reporting movement in the pelvis.

One of the most dynamic lumbar-pelvic motions assessed with different Cardan sequences considered the lower limb joints and the pelvis during the instep soccer kick[10]. In the 2010 study,

Lees et al concluded a sequence of either XYZ or XZY were preferred. Flexion/extension was found to be the angle most easily described across sequences, while abduction/adduction and internal/external rotations possessed more uncertainties, with variability in shape and offset magnitudes with different sequences.

Systematic assessment of calculation methods on a diverse range of dynamic, coupled motions is needed across all parts of the human body. This study aims to determine the effect of Cardan rotation sequences on lumbar spine motions during symmetric and asymmetric (coupled) lifting tasks.

2.2: Methods

2.2.a: Subjects

Twenty-six subjects participated in this study (25 ± 5 years old, 24.7 ± 3.6 BMI, 10 women, 16 men) with the approval from the Human Subjects Committee from the University of Kansas, Lawrence, Kansas. Twenty-three subjects were included in the analysis (25 ± 5 years old, 24.9 ± 3.9 BMI, 9 women, 14 men). Subjects were recruited using fliers, email, and word-of-mouth. All subjects completed a consent form, and a medical and work history questionnaire. Subjects who had a history of lower back pain and/or cardiovascular disease were excluded from the study.

2.2.b: Experimental Setup

Electromagnetic motion sensors (MotionStar, Ascension Technology, VT) attached to custom clips (Fig. 2.2) were taped on the skin surface to the subject's manubrium, and T10 and S1 vertebrae using 2" wide Cover Roll Stretch Bandage (BSN Medical, Hamburg, Germany). The skin surface was cleaned of skin oil and sweat with alcohol wipes prior to sensor attachment. Excessive hair growth at the attachment site was removed with a razor and shaving cream. A

Velcro waistband was also added to support and provide tension relief to the sensor cords. Position and orientation data were collected using the Motion Monitor system (Innsport, IL) at 100Hz. As reported by the manufacturer, these sensors have a resolution of 0.08 cm and 0.1° and an RMS accuracy of 0.76 cm and 0.5°.

Fig. 2.2: Custom 3D-printed clips holding electromagnetic sensors used in study

To locate the T10 vertebra, the C7 spinous process was first located via palpation at the base of the neck with the subject's chin lowered to their chest. Subsequent vertebrae were found by palpation until T10 was located. Subjects were asked to wrap their arms around themselves as though they were hugging themselves to help with vertebrae palpation. The S1 vertebra was located by palpating for the tops of the ilium to find the sacroiliac joints (Fig. 2.3). The S1 vertebra is found between the sacroiliac joints.

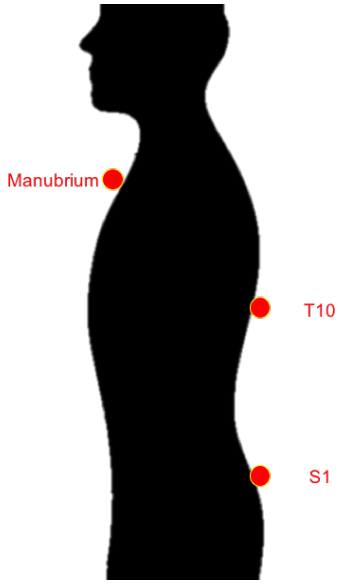

Fig. 2.3: Placement of electromagnetic motion sensors on the manubrium and the T10 and S1 vertebrae

After the sensors were attached, subjects were familiarized with moving with the attachments while the range between the sensors and the receiver was checked. Continued secure sensor attachment was periodically checked throughout the experimental protocol. Subjects were asked to report if they felt the taped sensors shift or become unattached. At that point, the sensor would be reattached with new or additional tape and the task motion would be recollected.

2.2.c: Experimental Protocol

First an investigator performed with participants and simultaneously guided them verbally through three tasks to establish the range of motion of a subject's lumbar posture (Fig. 2.4), lateral left-right bending (Fig. 2.5), and axial left-right twist (Fig. 2.6). Each motion was performed three times at five torso flexion angles: upright, 40°, 60°, and 80° (Fig. 2.7). Subjects were told to minimize knee bend in standing tasks.

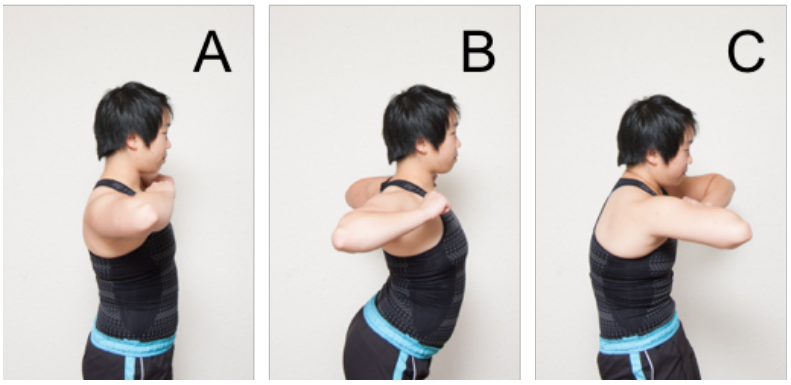

Fig. 2.4: Neutral upright lumbar posture (A), more lordotic lumbar posture (B), and more kyphotic lumbar posture (C)

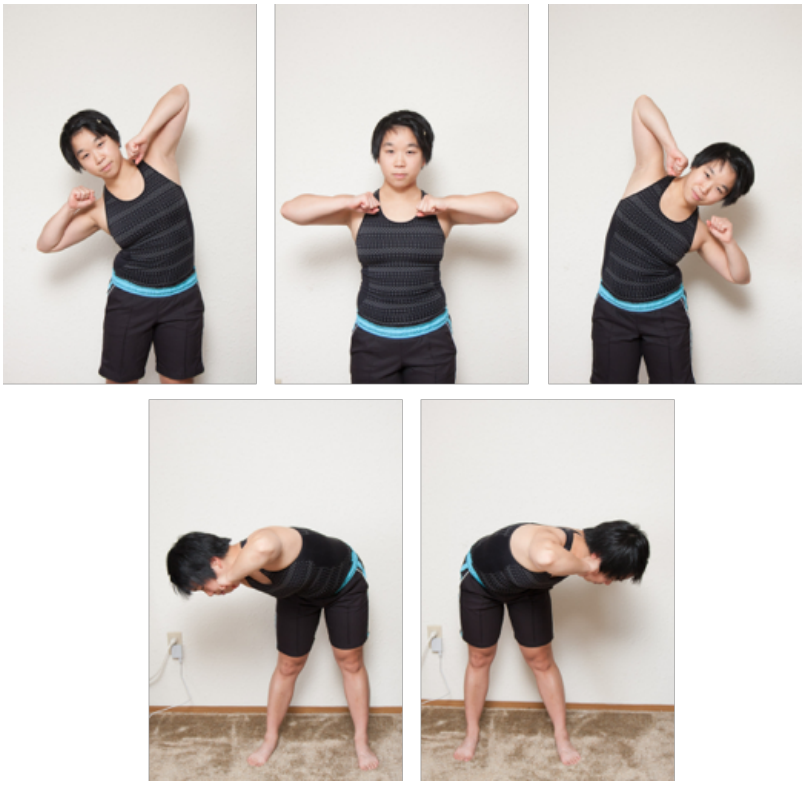

Fig. 2.5: Lateral right and lateral left rotations in an upright position and in a flexed position

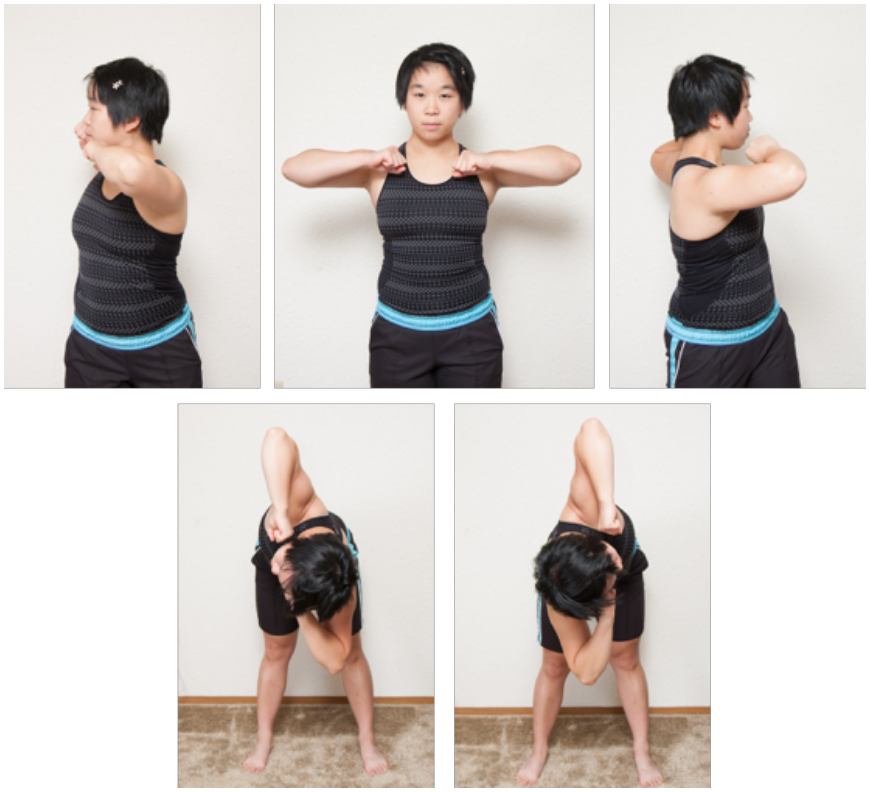

Fig. 2.6: Axial right and axial left rotations in an upright position and in a flexed position

Fig. 2.7: Torso flexion for range-of-motion tasks: upright, 20°, 40°, 60°, and 80°

Subjects then performed one minute of cyclic lifting of a 12lb crate in six setups (Fig. 2.8-2.11): (1) straight-legged lifting of a crate on the ground directly in front of the subject, (2) standing straight-legged lifting of a crate on the ground 45° to the left of the subject, (3) standing straight-legged lifting of a crate on the ground 45° to the right of the subject, (4) seated lifting of

a crate on a box directly in front of the subject, (5) seated lifting of a crate on a box 45° to the left of the subject, and (6) seated lifting of a crate on a box 45° to the right of the subject. The order of seated and standing tasks, and the directional order of the standing and seated tasks performed by the subjects were block randomized. In each task, a verbal count given by an investigator to a metronome sounding at 60bpm guided subjects through each cyclic lift. Subjects held the crate for two seconds at their waist, were given two second to move the crate to the ground or box, allowed the crate to rest for two seconds, and were given two seconds to lift the crate to their waist. For standing tasks, subjects were asked to place the crate along the taped direction at a distance that was the further away but still comfortable for them to consistently perform the lift cycle. Similarly, for the seated tasks, subjects were asked to determine the placement of the box the crate would be set on at the furthest comfortable distance.

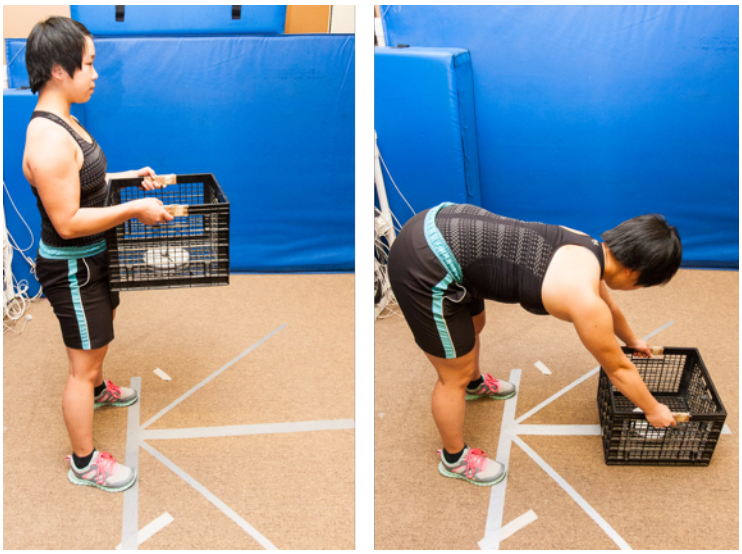

Fig. 2.8: Standing straight-legged lifting of crate on ground directly in front of subject. Participants are asked to select a distance along the taped line that is furthest away but still comfortable for them to repeatedly place and lift the crate from.

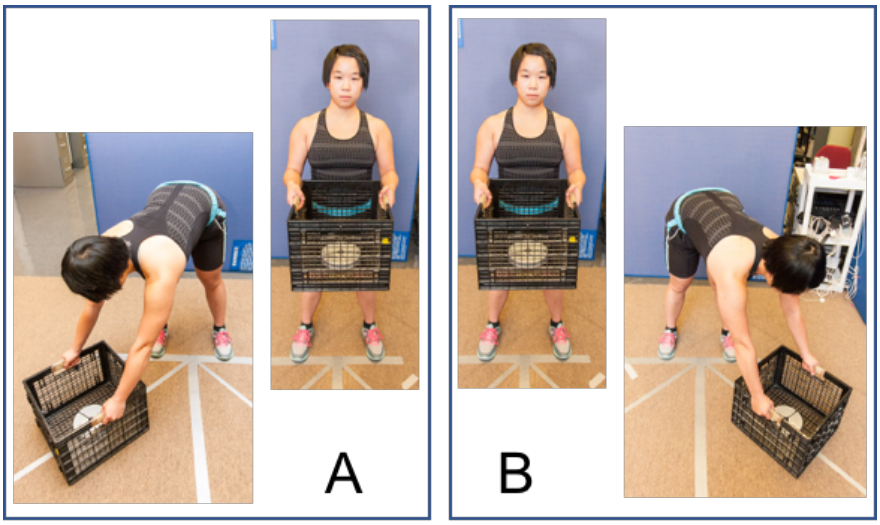

Fig. 2.9: Two end positions for standing straight-legged lifting of crate on ground from 45° to the right (A) and 45° to the left (B). Participants are asked to select a distance along the taped line that is furthest away but still comfortable for them to repeatedly place and lift the crate from.

Fig. 2.10: Two end positions for cyclic seated lifting of crate directly in front of subject

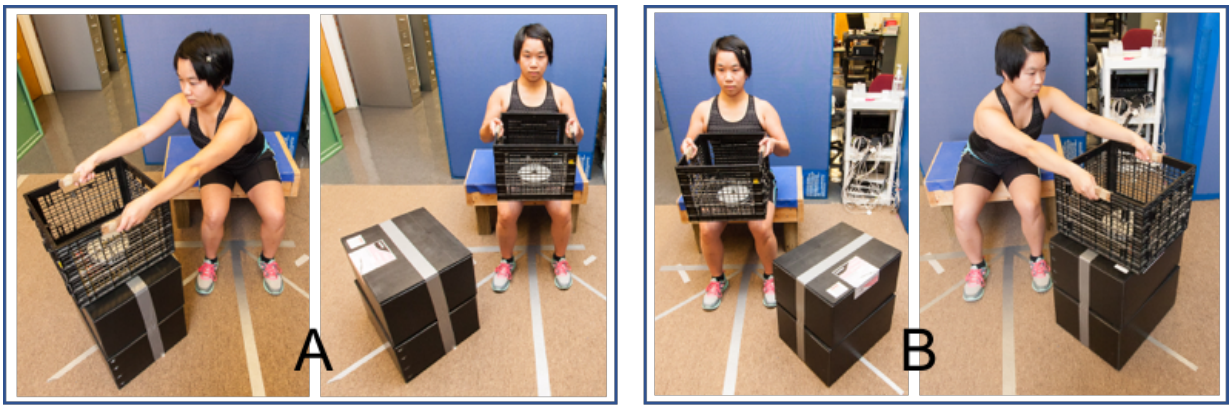

Fig. 2.11: Two end positions for cyclic seated lifting of crate 45° to the right (A) and 45° to the left (B)

2.2.d: Analysis

Data processing was performed using MATLAB. Orientation of individual sensor axes was checked through animation of the raw data. Sensor positions are given with an x, y, and z component. The distance from the receiver is calculated as sum of x^2, y^2, and z^2. Next, if any of the sensor positions were found to be greater than 1, which would indicate the sensor has moved beyond the receiver's range of one meter, the data point would be changed to not-a-number (NaN). For instances in which the sensor's position was recorded at (0,0,0) – which would mean the sensor is too close the receiver (within 10cm) – those data points would also be changed to NaN.

The raw orientation data of each sensor is returned for each collected data point as a 3x3 rotation matrix. For all collected data, this rotation matrix was adjusted to set the sensor's initial orientation as facing the positive global X direction (so pelvic torsion = 0). This adjustment was made by multiplying the raw quaternion rotation matrix with the matrix [0 0 -1; 0 -1 0; -1 0 0], which would yield an identity matrix for a subject standing upright:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} R'_{11} & R'_{12} & R'_{13} \\ R'_{21} & R'_{22} & R'_{23} \\ R'_{31} & R'_{32} & R'_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textit{if subject is standing upright at the collected data point}$$

Another rotation adjustment is applied to place the sensors relative to the direction the subject is facing. This is done by adjusting the manubrium and T10 sensors to the upright z-axis orientation of the S1 sensor through the following matrix multiplication, where ∅ is the S1 z-axis Euler angle set to the direction the subject is facing and the R' matrix is the rotation matrix at each collected time point adjusted to face global X:

$$\begin{bmatrix} \cos\emptyset & -\sin\emptyset & 0 \\ \sin\emptyset & \cos\emptyset & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} R'_{11} & R'_{12} & R'_{13} \\ R'_{21} & R'_{22} & R'_{23} \\ R'_{31} & R'_{32} & R'_{33} \end{bmatrix}$$

For this study, all subjects were facing global X and $\varnothing = 0$, so the R' matrix does not change with the multiplication with an identity matrix. After these orientation adjustments were made, for each Cardan sequence, Euler angles were calculated for the three sensors using the MATLAB's dcm2angle function and converted from radians to degrees.

The axis for a virtual torso sensor was constructed from the position of the manubrium, T10, and S1 sensors through a series of vector construction and cross products shown in Fig. 2.12. Similar to the adjustments to the physical manubrium and vertebra sensors, a rotation correction with the matrix [-1 0 0; 0 1 0; 0 0 1] was applied to the torso sensor to achieve a sensor orientation facing the global X direction, and then adjusted relative to the transverse plane orientation of the S1 sensor.

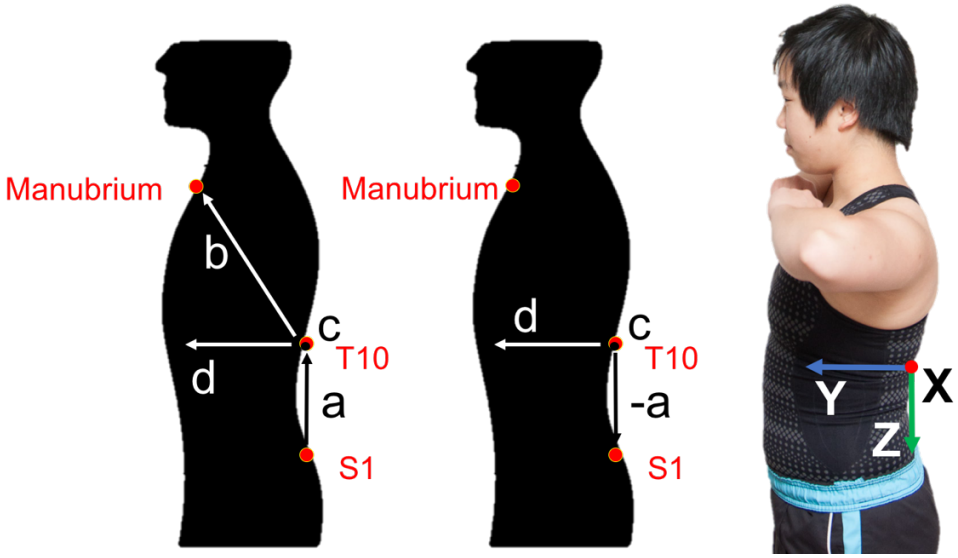

Fig. 2.12: Construction of the torso axis: vector a (T10 – S1) is crossed with vector b (Man – T10) to create vector c ; vector a and vector c are crossed to create vector d ; vectors –a, c, and d are normalized to form the final torso coordinate axis, where vector c is the x-axis, vector d is the y-axis, and vector -a is the z-axis.

Four angles were calculated: (1) trunk flexion, (2) lumbar flexion, (3) lumbar lateral rotation, and (4) lumbar axial rotation. Trunk flexion was calculated as the rotation about the x-axis of the virtual torso sensor. Lumbar flexion was calculated as the difference between the x-axis rotation of the T10 sensor and the x-axis rotation of the S1 sensor. Lumbar axial rotation was calculated as the difference between the z-axis rotation of the T10 sensor and the z-axis rotation of the S1 sensor. Lumbar lateral rotation was calculated as the difference between the y-axis rotation of the T10 sensor and the y-axis rotation of the S1 sensor. This calculation was done for the six Cardan sequences.

To understand the effect of Cardan sequence selection, plots comparing Cardan sequence values for each calculated angle were made through further data analysis (Fig. 2.13). First, for each subject, one lift cycle was selected from the second third of the trial. The lift cycle was selected based on trunk flexion, starting from the most flexed position in a lift task. Interpolation refitted the lift cycle into 100 data points. The mean and standard deviation across subject was taken at each point of the lift cycle for the four Euler angles (trunk flexion, lumbar flexion, lumbar lateral rotation, and lumbar axial rotation). The means were plotted to compare Euler angle values across sequences.

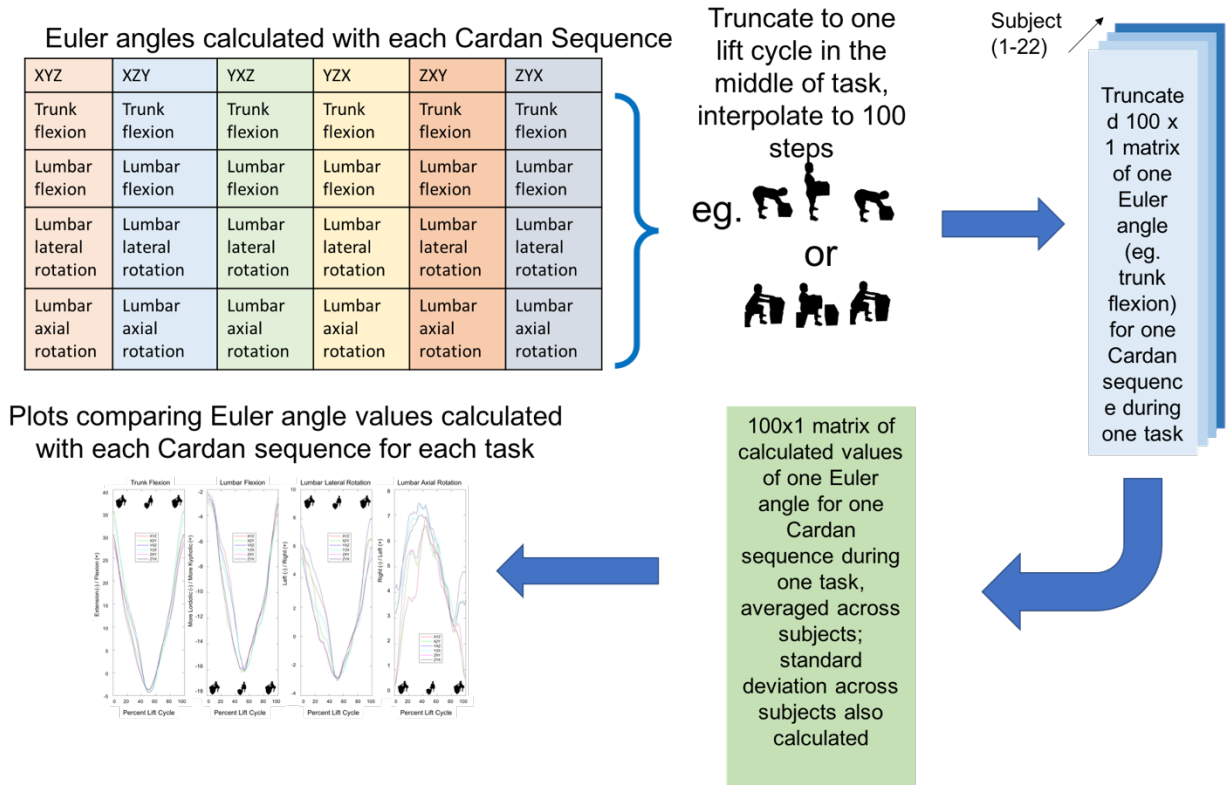

Fig. 2.13: Map of steps taken in data analysis after Euler angles for each Cardan sequence are calculated.

2.3: Results

2.3.a: Centered Tasks

During centered standing crate lifting, all Cardan sequences were able to produce Euler angles in an appropriate range of motion with the exception of YXZ and ZXY (Fig. 2.14). In this task, YXZ and ZXY show gimbal lock artifacts and artifacts of large magnitude in lumbar lateral rotation and lumbar axial rotation. In deep flexion, YXZ and ZXY produced trunk and lumbar flexion values slightly lower than the other four sequences.

During centered seated crate lifting, all Cardan sequences were able to describe the expected path of motion for the four calculated spine angles (Fig. 2.15). Subgroups of sequences yielding similar paths of motion can be seen in lumbar flexion and lumbar lateral rotation.

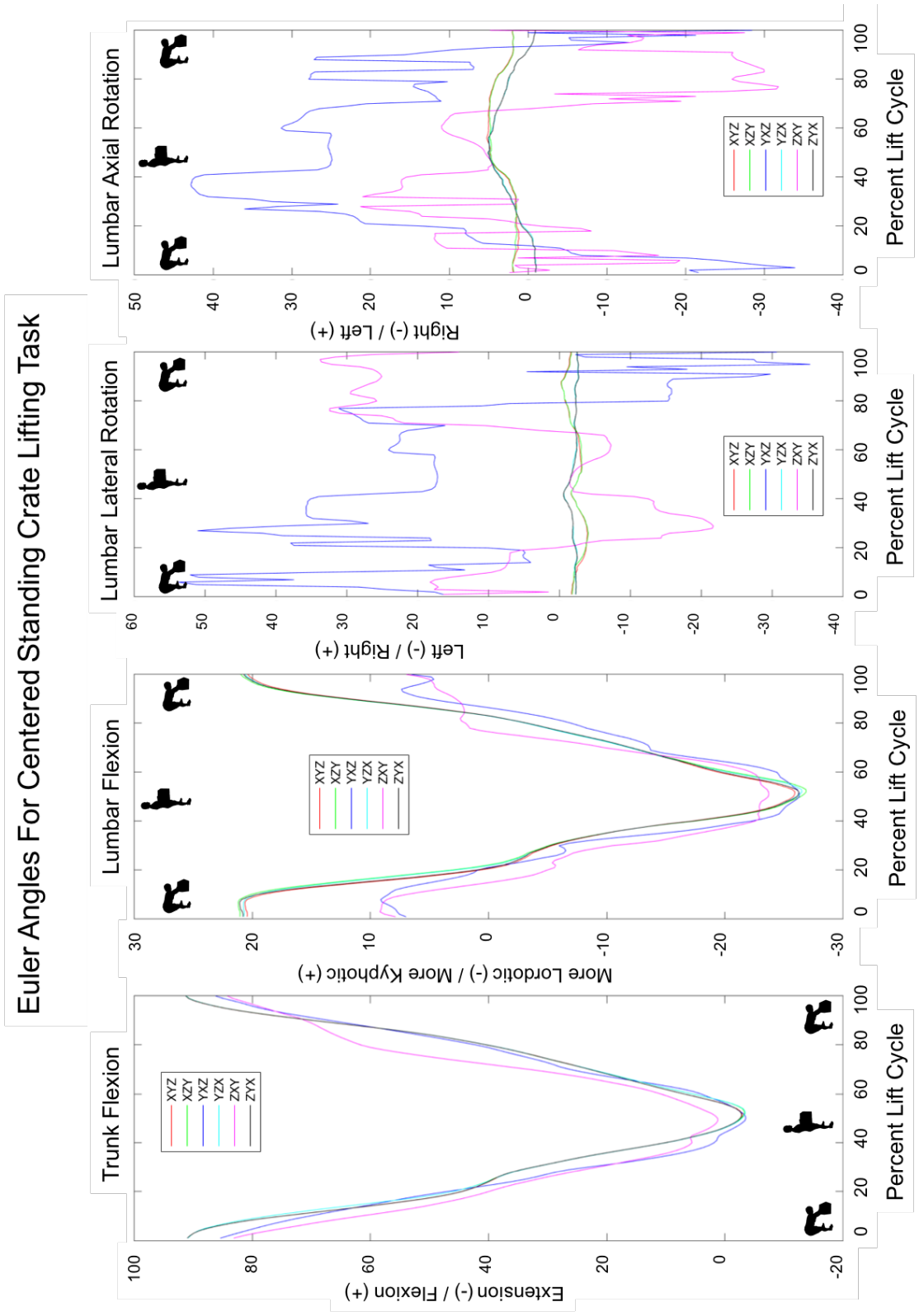

Fig. 2.14: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the standing crate lifting task without rotation. 0% of the lift cycle corresponds to the subject bent over with the crate on the ground. The crate is lifted up, and returned to the ground by the end of the lift cycle (100%).

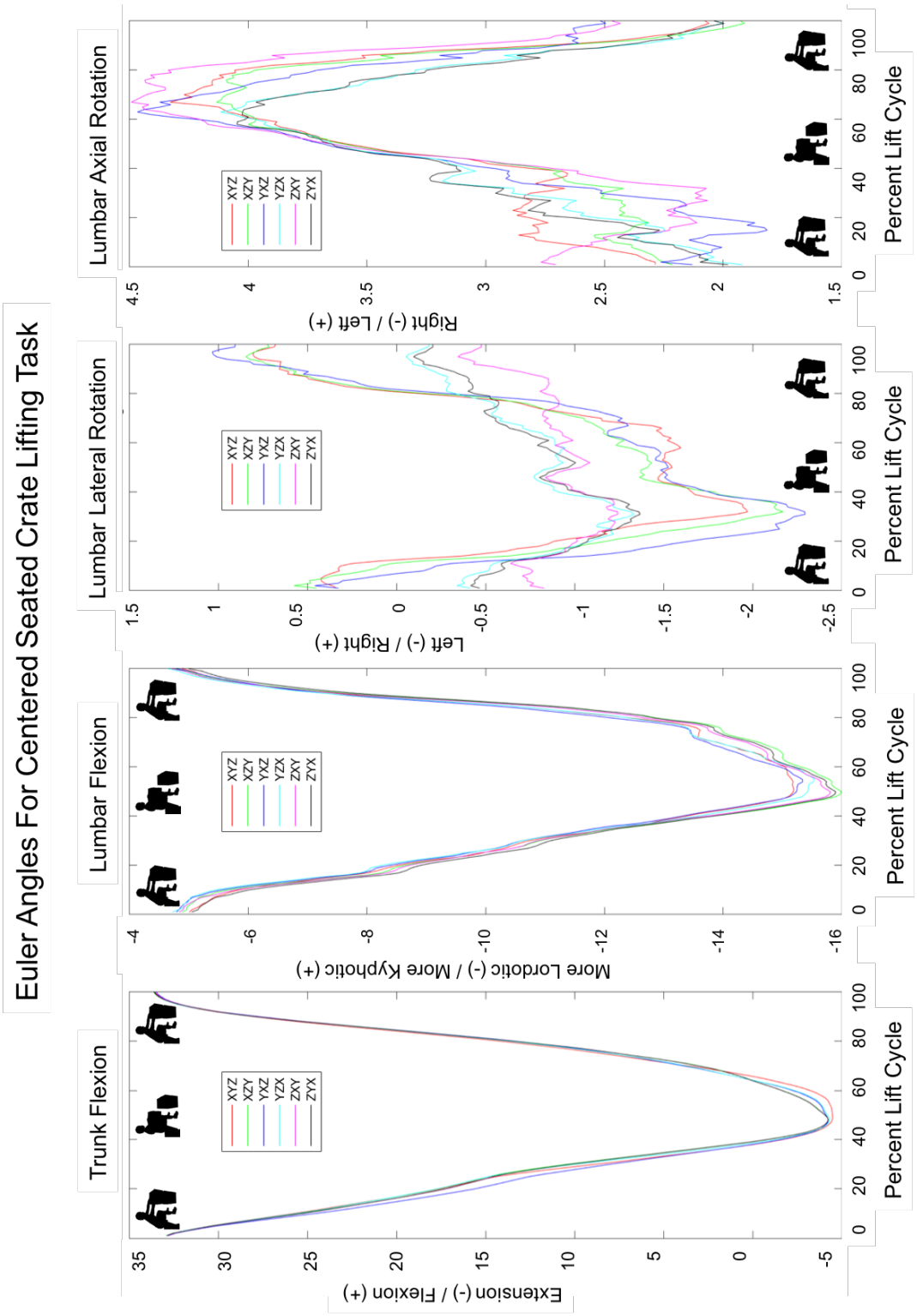

Fig. 2.15: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the cyclic task lifting a crate to and from a box located directly in front of the subject seated on a bench. 0% of the lift cycle corresponds to the subject leaning forward with arms extended to grasp the crate on top of the box. The crate is pulled in above the subject's lap with the subject coming to an upright seated position. The lift cycle ends when the crate is returned to the top of the box in front of the subject (100%).

2.3.b: Asymmetric Tasks

During standing asymmetric tasks, we continue to see gimbal lock artifacts and artifacts of large magnitude in lumbar lateral rotation and lumbar axial rotation with YXZ and ZXY (Fig. 2.16-2.17). The lower values of trunk and lumbar flexion for YXZ and ZXY in deep flexion persists. Artifact peaks and troughs are also seen in trunk and lumbar flexion. In lumbar axial rotation, YZX and ZYX appear to describe the asymmetric rotation opposite of the expected path: for 45° left, YZX and ZYX show a more right-side rotation when subjects should be twisted to the left, and vice versa for 45° right.

During seated asymmetric tasks, subgroups of sequences for lumbar axial rotation are observed (Fig. 2.18-2.19). The sequence subgroups for lumbar flexion and lumbar lateral rotation seen in the centered seated task are also seen in the asymmetric tasks. These subgroups are summarized in Table 2. One subgroup in lumbar axial rotation appear to describe the rotation opposite to the expected direction. This subgroup includes sequences YXZ, YZX, and ZYX.

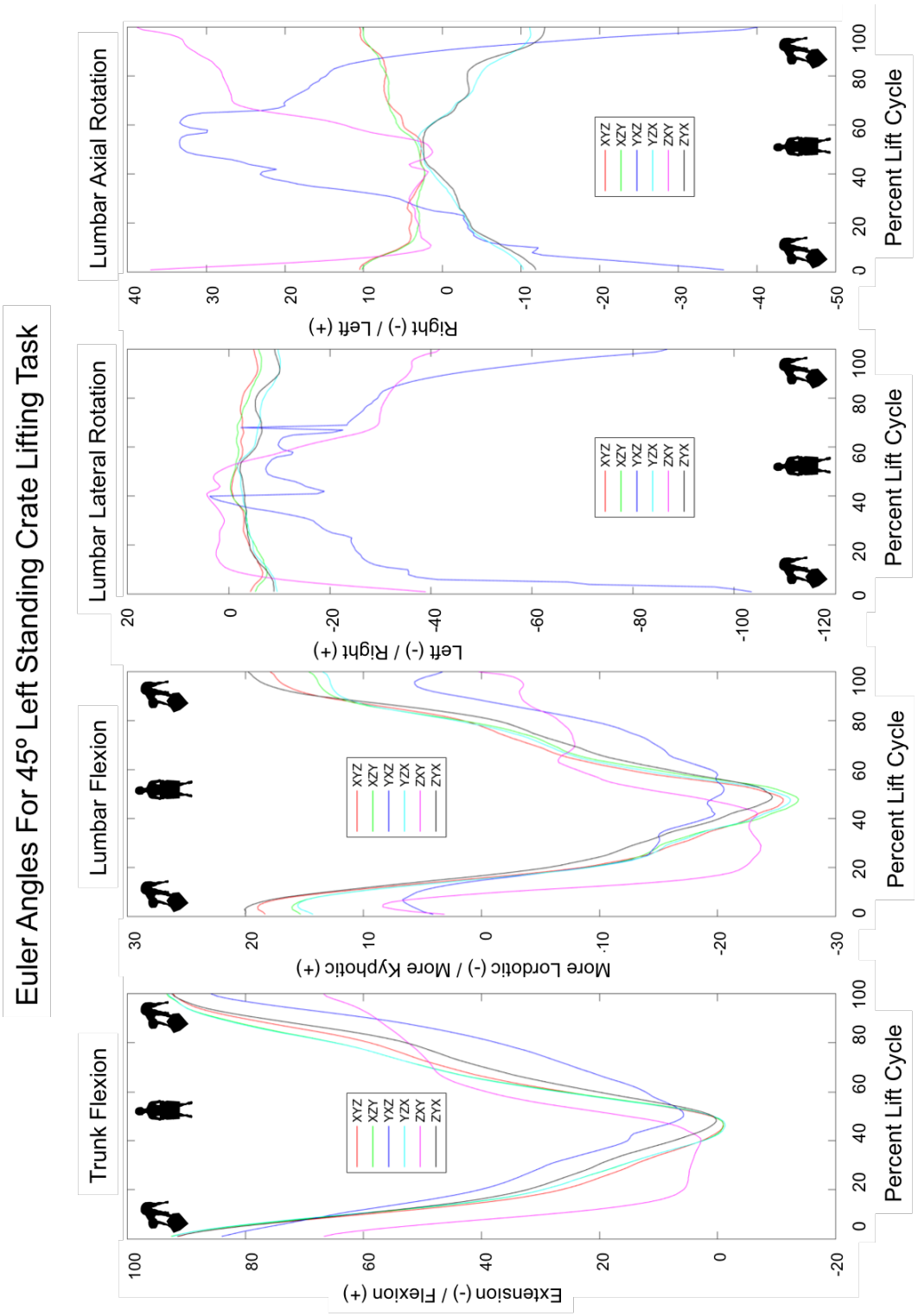

Fig. 2.16: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the cyclic task lifting a crate to and from a position 45° to the left of the center. 0% of the lift cycle corresponds to the subject bent over with the crate on the ground. The crate is lifted up, and returned to the ground by the end of the lift cycle (100%).

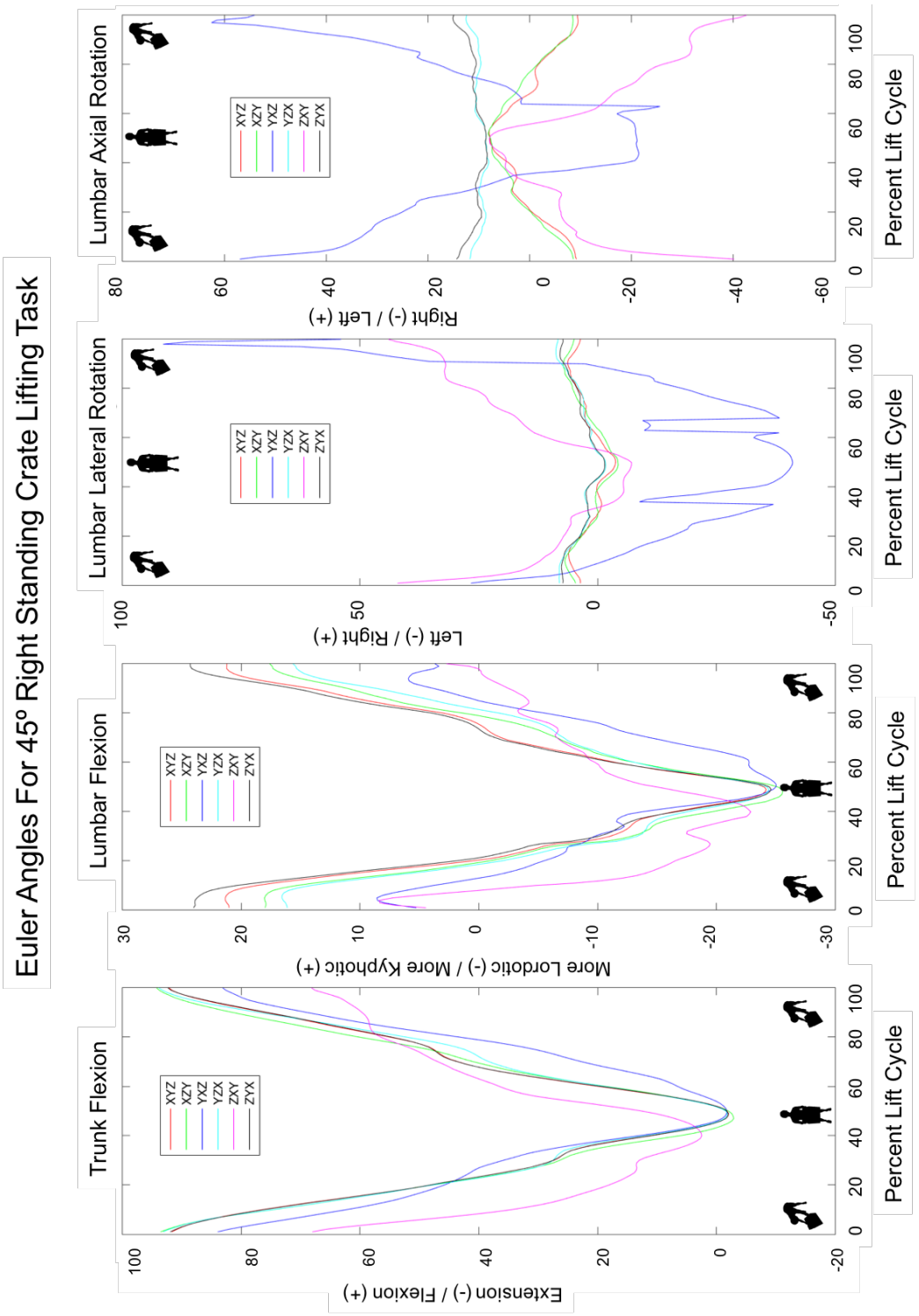

Fig. 2.17: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the cyclic task lifting a crate to and from a position 45° to the right of the center. 0% of the lift cycle corresponds to the subject bent over with the crate on the ground. The crate is lifted up, and returned to the ground by the end of the lift cycle (100%).

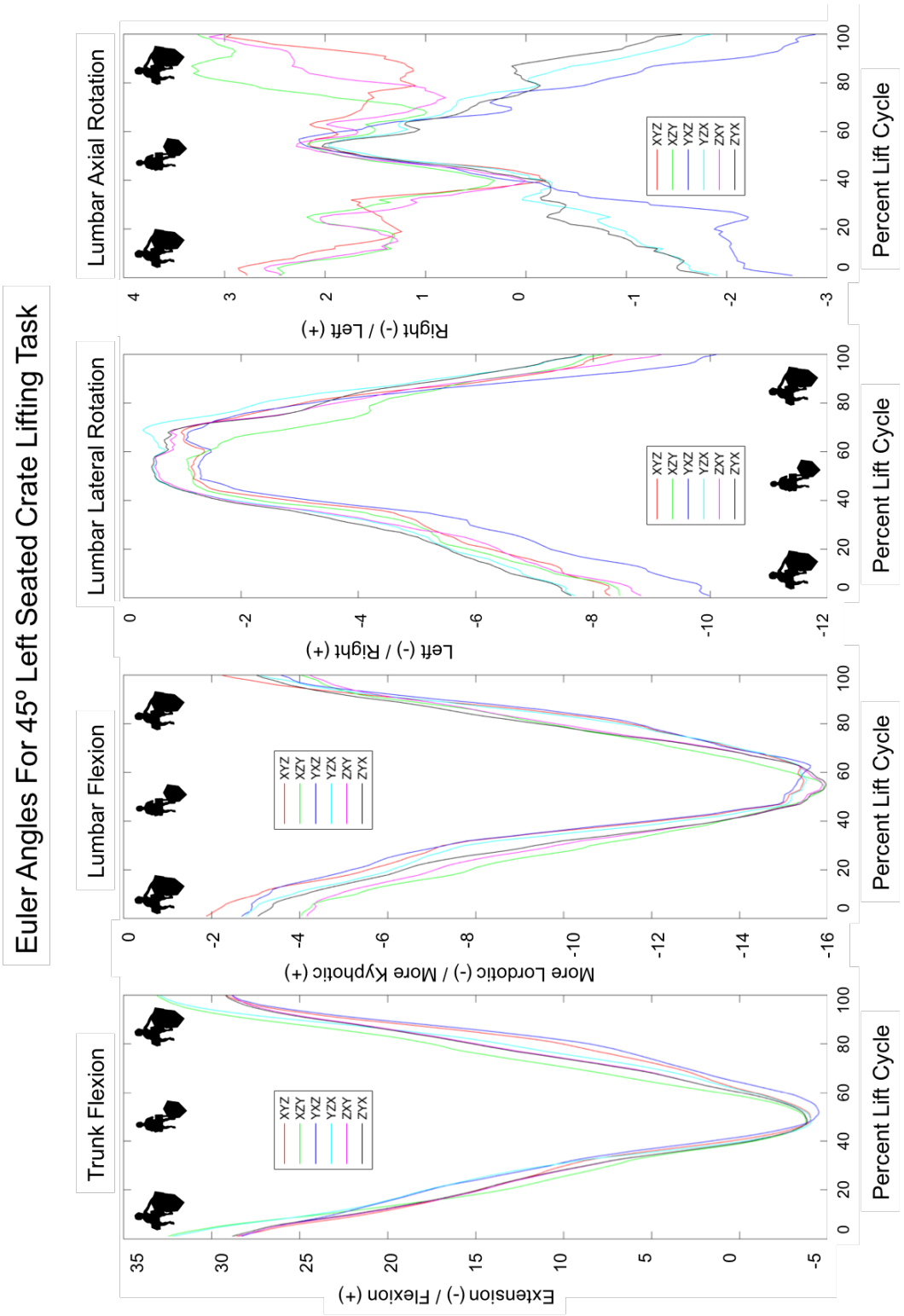

Fig. 2.18: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the cyclic task lifting a crate to and from a box located along a line 45° to the left of the subject, who is seated on a bench. 0% of the lift cycle corresponds to the subject leaning forward with arms extended to grasp the crate on top of the box. The crate is pulled in above the subject's lap with the subject coming to an upright seated position. The lift cycle ends when the crate is returned to the top of the box in front of the subject (100%).

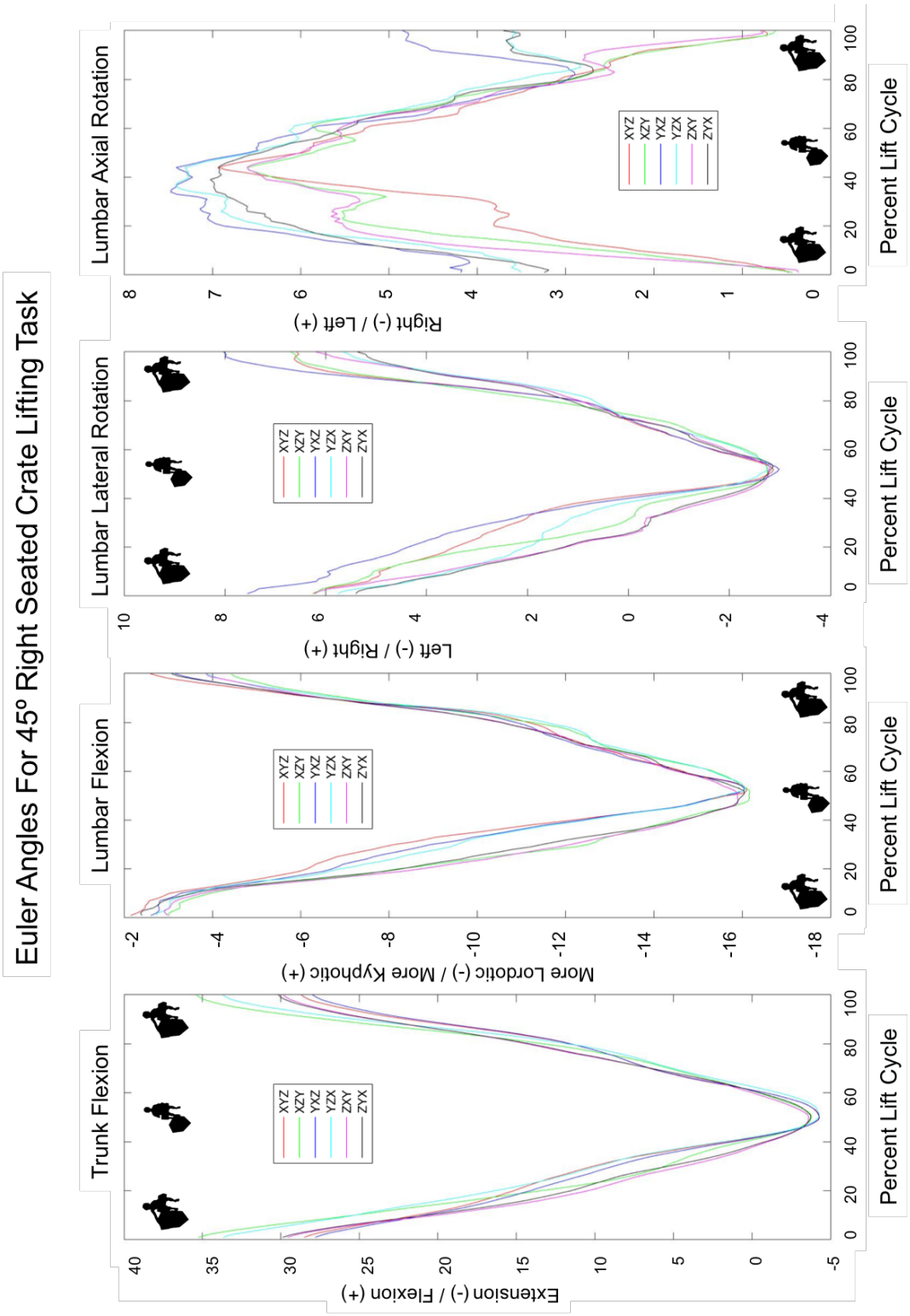

Fig. 2.19: Descriptions of four Euler angles (trunk flexion, lumbar posture, lumbar lateral bending, and lumbar torsion) from six Cardan sequences for one cycle of the cyclic task lifting a crate to and from a box located along a line 45° to the right of the subject, who is seated on a bench. 0% of the lift cycle corresponds to the subject leaning forward with arms extended to grasp the crate on top of the box. The crate is pulled in above the subject's lap with the subject coming to an upright seated position. The lift cycle ends when the crate is returned to the top of the box in front of the subject (100%).

2.3.c: Standing vs Seated Tasks

In all standing tasks, descriptions of lumbar lateral rotation and lumbar axial rotation were affected with gimbal lock and large increases. These averaged descriptions are magnitudes greater than physically possible, moving through a path of rapid and large increases contrary to the smooth continuous motions subjects performed. Subject plots of the T10 sensor (such as the ones given in Fig. 2.20) show that these descriptions come from gimbal lock increases in the y- and z-rotation angles when calculated with a sequence with X as the second rotation.

In all seated tasks, all Cardan sequences produced Euler angles within 10° for all four angles when averaged across 22 subjects (Fig. 2.14, 2.16, 2.17). Two subgroups of Cardan sequences producing similar values can be seen in lumbar flexion, lumbar lateral rotation, and lumbar axial rotation (Table 2).

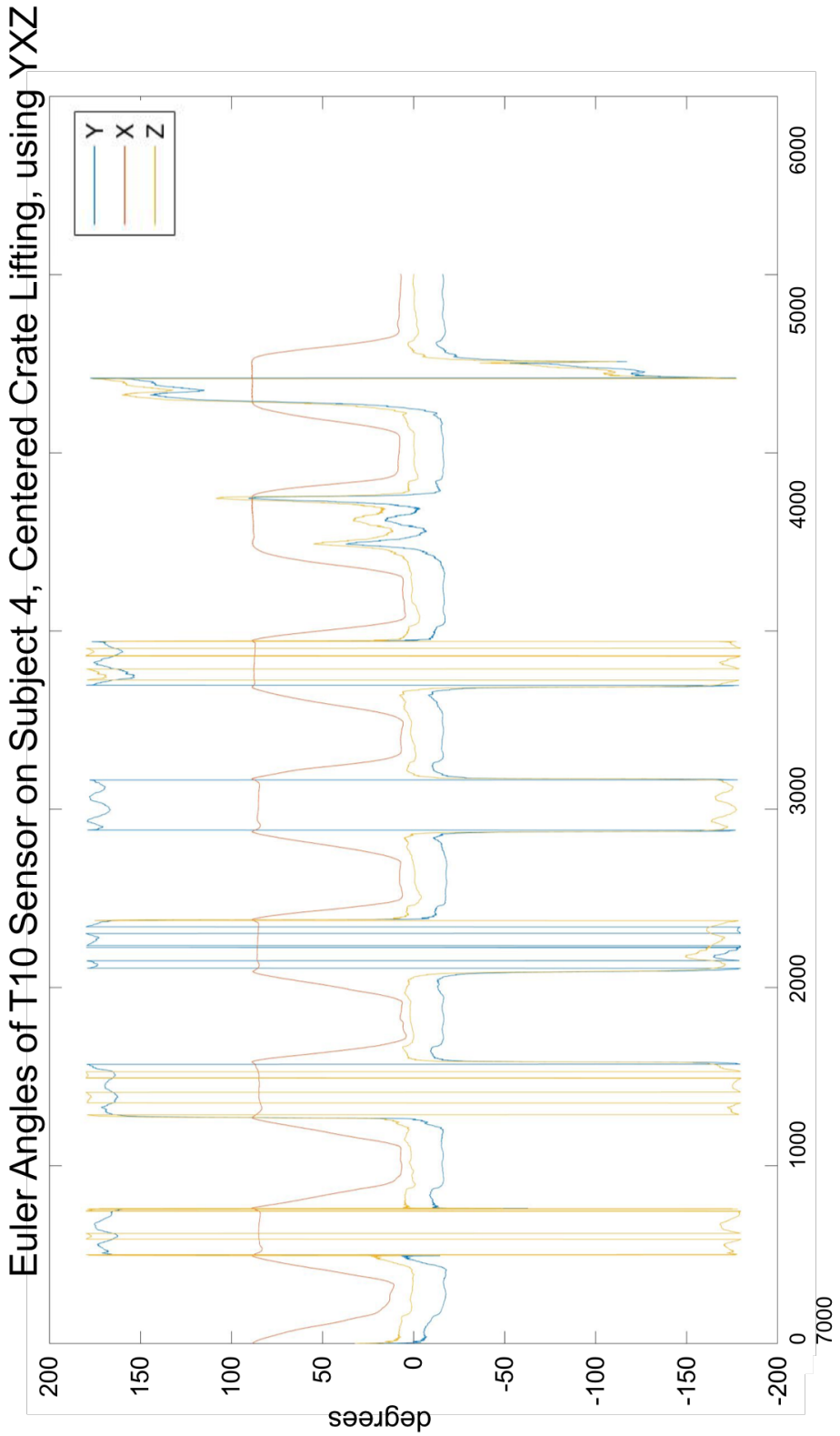

Fig. 2.20: Presence of gimbal lock in y- and z-rotation angles of the T10 Sensor when calculated with sequence YXZ for a single subject during centered crate lifting

Table 2: Cardan sequence subgroups producing similar Euler angle values in seated lifting tasks

	Subgroup 1	Subgroup 2
Lumbar Flexion	XYZ, YXZ, YZX	XZY, ZXY, ZYX
Lumbar Lateral Bending	XYZ, XZY, YXZ	YZX, ZXY, ZYX
Lumbar Axial Rotation	XYZ, XZY, ZXY	YXZ, YZX, ZYX

2.4: Discussion

2.4.a: Gimbal Lock in Standing Tasks

In standing tasks, Cardan sequences that placed the flexion axis (X) as the second rotation – YXZ and ZXY – produced Euler angles that were most divergent from the expected description. Placement of the largest angle in the second rotation step results in inflated values of lumbar lateral rotation and lumbar axial rotation, and introduction of artifact motion that did not occur in all calculated angles in trunk and lumbar flexion. A look at the effect of Cardan sequences on the Euler angles of the T10 and S1 sensors on individual subjects – which are then used to calculate lumbar flexion, lumbar lateral rotation, and lumbar axial rotation – show the inflated artifact motion resulting from gimbal-lock trends in the y- and z-axis Euler angles for YXZ and ZXY when the x-rotation angle is near or at 90°. The presence of gimbal locks in such instances can be explained mathematically. Using the sequence YXZ as an example,

$$R_{yxz} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

When the x-rotation angle, $\alpha = 90°$, this rotation matrix becomes:

$$R_{yxz} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Which multiples out and, with trigonometric identities, becomes the following mathematical singularity, where no matter how β or γ changes, the second row and third column of R_{xyz} remain unchanged:

$$R_{yxz} = \begin{bmatrix} \cos\beta\cos\gamma - \sin\beta\sin\gamma & -\cos\beta\sin\gamma - \sin\beta\cos\gamma & 0 \\ 0 & 0 & -1 \\ \sin\beta\cos\gamma - \cos\beta\sin\gamma & -\sin\beta\sin\gamma - \cos\beta\cos\gamma & 0 \end{bmatrix}$$

$$R_{yxz} = \begin{bmatrix} \cos(\beta+\gamma) & -\sin(\beta+\gamma) & 0 \\ 0 & 0 & -1 \\ \sin(\beta-\gamma) & -\cos(\beta+\gamma) & 0 \end{bmatrix}$$

Furthermore, the MATLAB function used to calculate Euler angles for each sensor based on Cardan sequence, dcm2angle, may contribute to the challenge of obtaining reasonable y- and z-axis rotation values for YXZ and ZXY. In the Mathworks documentation for the function, dcm2angle produces a set of three rotation angles based on a direction cosine matrix:

[r1 r2 r3] = dcm2angle (n, s)

where n is the direction cosine matrix, s is the specified rotation sequence. A default limitation of the function when the specified rotation sequence is Cardan is that r1 and r3 will lie between ±180° and r2 will lie between ±90°. When the flexion axis (X) is the second rotation, r2, but the executed motion involves flexion that may be greater than 90°, as in the standing tasks, this may lead to difficulty producing correct values for the other two rotation directions, as well as the more conservative values of trunk and lumbar flexion.

While a metronome and verbal count from an investigator guided subjects through the tasks, the rate at which subjects moved into and out of a lift cycle is slightly varied. This variability in combination with the gimbal-lock trends in these two Cardan sequences leads to a messy average. This variability is not exacerbated by a gimbal-lock trend for the other sequences, and as such, the averages tell a clearer description of the sequences' effect.

2.4.b: Sequence Subgroups in Seated Tasks

Most notable in the seated tasks are the subgroups of sequences producing closely similar Euler angles for lumbar flexion, lumbar lateral rotation, and lumbar axial rotation (Table 1). These subgroups are consistent with the findings and mathematical proof presented by Crawford et al which approximated the subgroupings to hold for small angles (approximated at 30°)[8]. As an average, subjects flexed less than 35° in seated tasks (Fig. 2.15, 2.18, 2.19).

2.4.c: Holistic Assessment

Looking at the six lifting tasks, there are two holistic assessments related to the choice in Cardan sequence. The first is the robustness of the calculated spine angle against sequence selection. Of this, trunk flexion shows the most consistency and least amount of change between sequences. Following trunk flexion, an order of spine rotation robustness could be given as: lumbar flexion, lumbar lateral rotation, and lastly, lumbar axial rotation. Both trunk and lumbar flexion are affected in instances of deep flexion, with lumbar flexion showing greater differences than trunk flexion. However, both are still able to remain in an appropriate range of motion and describe the motion taken. For two of the six sequences in standing tasks, both lumbar lateral rotation and lumbar axial rotation are largely affected by gimbal-like artifacts; however, lumbar axial rotation showed additional inconsistencies in reporting the correct direction of motion with different Cardan sequences.

The second holistic assessment is the robustness of a Cardan sequence's ability to correctly describe the motions performed. Across all six lifting tasks, the sequences XYZ and XZY proved resilient in producing Euler angles descriptive of the motions performed, whether they involved deep flexion or coupled, asymmetric motion. In comparison, YXZ and ZXY failed in describing

lumbar lateral rotation and lumbar axial rotation for standing tasks. YZX and ZYX, along with YZX, showed difficulty correctly indicating the direction of lumbar axial rotation. The robustness of XYZ and XZY reflect the sequence-selection logic proposed by Nowinski et al for using the sequence placing the axis of primary motion as the first in the sequence, followed by the coupled axes[12]. Previous studies have also noted the ability for XYZ to be based on anatomical definitions and translated to clinical evaluations[15-17].

The greatest differences due to Cardan sequence for the four calculated spine angles is seen at the start and end of the lift cycle in standing tasks, where trunk flexion was greatest, and as subjects move into and out of deep flexion. This finding corroborates with previous studies showing increases in differences caused by Cardan sequence selection in deep trunk flexion[8,13].

2.4.d: Limitations and Next Steps

Some limitations to this study included experimental setup. Sensor attachment is one possible source of error in this study. Sensors were placed on the external body and attached using tape rather than an in-bone rigid attachment to vertebra or manubrium. As such, there are instances in which the sensors may have shifted relative to the spine due to skin motion. Some subjects had defined back musculature that created a groove with the skin superficial to vertebrae and the adjacent spinal erector musculature. The groove made it difficult for the sensor in the custom clip to make direct contact with the skin. However, unless a sensor is attached rigidly to the vertebra, all the sensors in use currently present a similar problem with motion artifact due to the sensor-skin interface. Such a rigid attachment to the vertebra would be much more invasive for subjects. With the available equipment, the setup and protocol minimized such errors. Multiple tapes were tested before the Cover Roll Stretch tape was selected for use. Nonetheless, improvements to this

study would seek a sensor attachment method that further minimized possible shifting or detachment issues.

The collection range of the Motion Monitor system used was limited to a distance of one meter from the receiver. This created some problems for taller subjects, especially when performing standing lifting tasks. In such instances, a tall subject may move outside or to the edge of the receiver's range, leading to a loss of collected data at those moments. To minimize this problem, subjects practiced a lifting motion while an investigator checked the sensors' range through Motion Monitor's Real Time Animation. This study would be improved using sensors with a larger range. The motions performed in this study were not hindered by the cables attached to the sensors. However, for assessing more dynamic motions in future studies, use of sensors that would not hinder or alter the motion is recommended.

Next steps to understanding the effect of Euler rotation sequences and other calculation methods on human motion include examination of more dynamic motions across multiple planes. Other commonly used calculation methods include the helical/screw axis, and Joint Coordinate System (JCS). This study looked at three directions of lifting in two positions. Future studies should begin to apply a similar systematic analysis for a variety of dynamic motions in other joints. In addition to a limited number of studies on dynamic motions, further examination of rotations outside of the sagittal plane is needed. Lees, et al noted in the greatest uncertainties appeared to be in the y- and z-axis Euler angles (lateral and axial rotations) especially at the beginning of the in-step soccer kick[10]. This is also seen in this study, as the y- and z-axis Euler angles were most affected by sequence selection.

With increased clarity on the accuracy and suitability of kinematic calculation methods, investigation of representations of the lumbar range of motion should be pursued for different

population groups. A strong motivation for lumbar spine research is the prevalence of low back pain. Having a robust and reliable method of measuring dynamic motions in the lumbar spine will allow researchers to measure the differences in lumbar spine motion and movement patterns in different populations, such as in those affected with low back pain, and those without.

2.5: Conclusions

Based on the results of this study, we would recommend either the XYZ or XZY sequence when assessing motions in the lumbar spine during lifting tasks and activities of similar motion ranges. YXZ and ZXY resulted in inflated values and artificial artifact motions in standing tasks where deep trunk flexion occurred. YZX and ZYX show difficulty reporting expected direction of motion in lumbar axial rotation. As such, we do not recommend using these sequences for the lumbar spine in activities of similar motion ranges. In agreement with previous lumbar-pelvic studies, Cardan sequence selection did not greatly affect calculated values for motions where trunk flexion does not exceed 45°.

Continued study of the effect of Cardan sequence selection on complex, dynamic motions in various joints are needed to develop a clear understanding of kinematics measurement methods. Additionally, development of motion and orientation tracking sensors independent of a receiver's location and line-of-sight would advance biomechanics research.

2.6: References

1. Arjmand N, Shirazi-Adl A. Biomechanics of changes in lumbar posture in static lifting. Spine (Phila Pa 1976) 2005;30(23):2637-48.
2. Dolan P, Adams MA. Repetitive lifting tasks fatigue the back muscles and increase the bending moment acting on the lumbar spine. J Biomech 1998;31(8):713-21.
3. Fujii R, Sakaura H, Mukai Y, Hosono N, Ishii T, Iwasaki M, Yoshikawa H, Sugamoto K. Kinematics of the lumbar spine in trunk rotation: in vivo three-dimensional analysis using magnetic resonance imaging. Eur Spine J 2007;16(11):1867-74.
4. Li G, Wang S, Passias P, Xia Q, Li G, Wood K. Segmental in vivo vertebral motion during functional human lumbar spine activities. Eur Spine J 2009;18(7):1013-21.
5. Shin JH, Wang S, Yao Q, Wood KB, Li G. Investigation of coupled bending of the lumbar spine during dynamic axial rotation of the body. Eur Spine J 2013;22(12):2671-7.
6. Wu G, Cavanagh PR. ISB recommendations for standardization in the reporting of kinematic data. J Biomech 1995;28(10):1257-61.
7. Baker R. Pelvic angles: a mathematically rigorous definition which is consistent with a conventional clinical understanding of the terms. Gait Posture 2001;13(1):1-6.
8. Crawford NR, Yamaguchi, G.T., Dickman, C.A. Methods for determining spinal flexion/extension, lateral bending, and axial rotation from marker coordinate data: Analysis and refinement. Human Movement Science 1996;15:55-78.
9. Karduna AR, McClure PW, Michener LA. Scapular kinematics: effects of altering the Euler angle sequence of rotations. J Biomech 2000;33(9):1063-8.
10. Lees A, Barton G, Robinson M. The influence of Cardan rotation sequence on angular orientation data for the lower limb in the soccer kick. J Sports Sci 2010;28(4):445-50.
11. Schache AG, Wrigley TV, Blanch PD, Starr R, Rath DA, Bennell KL. The effect of differing Cardan angle sequences on three dimensional lumbo-pelvic angular kinematics during running. Med Eng Phys 2001;23(7):493-501.
12. Nowinski GP, Visarius H, Nolte LP, Herkowitz HN. A biomechanical comparison of cervical laminaplasty and cervical laminectomy with progressive facetectomy. Spine (Phila Pa 1976) 1993;18(14):1995-2004.
13. McGill SM, Cholewicki J, Peach JP. Methodological considerations for using inductive sensors (3SPACE ISOTRAK) to monitor 3-D orthopaedic joint motion. Clin Biomech (Bristol, Avon) 1997;12(3):190-194.
14. Skalli W, Lavaste F, Descrimes JL. Quantification of three-dimensional vertebral rotations in scoliosis: what are the true values? Spine (Phila Pa 1976) 1995;20(5):546-53.
15. Cole GK, Nigg BM, Ronsky JL, Yeadon MR. Application of the joint coordinate system to three-dimensional joint attitude and movement representation: a standardization proposal. J Biomech Eng 1993;115(4A):344-9.
16. Grood ES, Suntay WJ. A joint coordinate system for the clinical description of three-dimensional motions: application to the knee. J Biomech Eng 1983;105(2):136-44.
17. Tupling SJ, Pierrynowski, M.R. Use of cardan angles to locate rigid bodies in three-dimensional space. Medical and biological engineering and computing 1987;25(5):527-532.

Chapter 3: Summary

3.1: Conclusions

The objective of this study was to assess the effect of Cardan sequence selection on Euler angles describing lumbar spine motion during cyclic lifting tasks moving within the sagittal plane and across multiple planes. This objective was addressed/investigated through a study collecting the kinematics of the lumbar spine in 22 human participants performing cyclic lifting tasks in two positions and three directions.

It was hypothesized that motion restricted to a single plane would be best represented by Cardan rotation sequences where the first rotation matched the plane of motion. In Chapter 2, two tasks examined occurred primarily in the sagittal plane: a centered standing crate lifting task and a centered seated crate lifting task. Between the two centered lifting tasks, XYZ, XZY, YZX, and ZYX were all able to describe the four calculated spine angles during lifting tasks moving primarily in the sagittal plane. Slight differences were seen between the sequence pairs XYZ and XZY, and YZX and ZYX in lumbar lateral rotation and lumbar axial rotation, but the four sequences were typically within 10° of each other. YXZ and ZXY gave poor representation of spine motion unless the greatest amount of motion in the sagittal plane was less than 45°. This is consistent with previous sagittal-plane motion studies concluding that Cardan sequence does not greatly affect calculated values for motions where trunk flexion does not exceed 45°.

It was hypothesized that motion occurring across multiple planes would be best represented by Cardan sequences with the planes of motion ordered from the one with the greatest motion to the least. In Chapter 2, four tasks were examined moving through multiple planes simultaneously: standing crate lifting at 45° to the left or right, and seated crate lifting at 45° to the left or right. In the four asymmetric tasks, the greatest amount of motion occurred in the sagittal plane about the

x-axis. In asymmetric standing tasks, the next plane was the z-axis, followed closely by motion about the y-axis. Range of motion about the y- and z-axes were similar, and even more so in seated tasks. From the asymmetric tasks, we can see that Cardan sequences ordered according to the magnitude of motion in each plane (in these cases, XYZ and XZY) *are* the best at representing motion across multiple planes. We see that sequences placing the axis with the greatest amount of motion as the second calculated rotation (in these cases, YXZ and ZXY) lead to poor representation of spine motion where trunk flexion is greater than 45°. Placing the axis with the greatest amount of motion in the third calculated rotation (in these cases, YZX and ZYX) lead to incorrect representation of direction for lumbar axial rotation.

In consideration of the assessment from Chapter 2, we would recommend XYZ or XZY as the sequences of choice when studying motions in the lumbar spine during lifting tasks and activities of similar motion ranges. We do not recommend the other four sequences in such cases: YZX and ZYX were not able to describe the expected direction of lumbar axial rotation; YXZ and ZXY resulted in inflated values and artificial artifact motions in standing tasks where deep trunk flexion occurred. Finally, when selecting a Cardan sequence for a study, the following considerations must be accounted for and clearly reported: the body part(s) being studied, the plane(s) the motion studied will occur in, the definition of local and global axes, and the assumptions and limitations of the motion capture system, and post-data collection analysis functions and corrections.

3.2: Future Works

This study and previous studies have observed that the optimal Cardan sequence(s) for one joint and type of motion may not be the most optimal for a different joint and motion. Continued study of the effect of Cardan sequence selection on complex, dynamic motions in a variety of joints

are needed to develop a clear and deep understanding of kinematics measurement methods. Studies with a systematic approach assessing kinematics measurement methods are needed. Additionally, development of motion and orientation tracking sensors independent of a receiver's location and camera line-of-sight would advance biomechanics research overall. Upon determination of a reliable and robust method, investigation of representations of the lumbar spine range of motion for different populations is recommended for furthering the understanding of different spine disorders.

To conclude: this study found XYZ and XZY to be the most robust Cardan sequences to describe lumbar spine motions during lifting activities. These are the recommended Cardan sequences given their robustness across tasks of varying ranges of motion across multiple planes.

Appendix A: MATLAB Code

Appendix A.1: Main body code

```
% Authors: Ednah Louie, S. Mukui Mutunga, Sara Wilson
% Revision date: 05-07-2018
% Program name: RotLift_controt14.m
% Purpose: This code calculates the mechanics of the lumbar
spine for six Cardan sequences, and prepares data matrices for
one lift cycle for further analysis. The steps are:
%     1. Import raw Motion Monitor data and initialize variables
%     2. Apply corrections for orientation to raw data for
analysis through called functions rotatecorr.m and
rotateback_cont.m
%     3. Calculate Euler angles for each sensor for a given
Cardan sequence through called functions MtoE2_dcm2angle.m and
LumFlexTor06.m
%     4. Build virtual torso sensor through called function
torsoMatrix6.m
%     5. Apply orientation corrections and calculating Euler
angles for virtual torso sensor
%     6. Determine trunk flexion, lumbar angle, lumbar lateral
bending, trunk lateral bending, and lumbar axial twist (torsion)
%     7. (optional) Plot T10 and S1 sensor Euler angles for
assessment
%     8. Create matrices of Euler angles for a single lift cycle
with use of called function StatCycle.m, averaging across
subjects
%     9. (optional) Create and saves figures for comparing
across Cardan sequences for Euler angles in a single lift cycle

% Motion Monitor Data by columns:
% 1 = time step (1000 Hz)
% 2-4 = sensor 2 xyz position at the manubrium
% 5-13 = quaternions for sensor 2
% 14-16 = sensor 3 xyz position at T10
% 17-25 = quaternions for sensor 3
% 26-28 = sensor 4 xyz position at S1
% 29-37 = quaternions for sensor 4
```

```
% ---- Variables ---- %

% isub = subject:
    % [2:6, 8:15, 17:25] are all subjects included in analysis;
    % [4,6,8,10:13,15,17,19:22,24,26] are weightlifters;
    % [2,3,5,9,14,18,23,25] are non-weightlifters


% idata = task:
    % 1 = kyphotic/lordotic ROM, 2 = lateral left/right ROM, 3 =
axial left/right ROM,
    % 4 = centered standing lifting, 5 = 45 L standing lifting,
6 = 45 R
    % standing lifting, 7 = centered seated lifing, 8 = 45 L
seated
    % lifting, 9 = 45 R seated lifting
% seqnum = sequence combination: 1 = XYZ, 2 = XZY, 3 = YXZ, 4 =
YZX, 5 =
    % ZXY, 6 = ZYX

close all
clear all
clc

path = ['/Volumes/LAB DRIVE/RotLift/MMFiles/']; % Define path
directory

for isub = [2:6, 8:15, 17:25]

    for idata = 4:9
        for seqnum = 1:6

            if isub < 10
                isubject = ['0' num2str(isub)];
            else
                isubject = num2str(isub);
            end


            % Define experience in weightlifting
            if isub == 4 || isub == 6 || isub == 8 || isub == 10
|| isub == 10|| isub == 11 || isub == 12 || isub == 13 || isub
== 15 || isub == 17 || isub == 19 || isub == 20 || isub == 21 ||
isub == 22 || isub == 24 || isub == 26
                exp = '(weightlifter)';
```

```
                status = 1;
            else
                exp = '(non-weightlifter)';
                status = 0;
            end


            % Correspond gender with subjects
            if isub == 2 || isub == 5 || isub == 6 || isub == 9
|| isub == 12 || isub == 13 || isub == 18 || isub == 21 || isub
== 25
                gen = 0;
            else
                gen = 1;
            end


            %%% Set up filenames for plots and data allocation
            if idata == 1
                filename = ['Sub' isubject '_NeutKL.exp'];
                fname = ['Subject ' isubject ' Kyphosis-
Lordosis'];
            elseif idata == 2
                filename = ['Sub' isubject '_LatLR.exp'];
                fname = ['Subject ' isubject ' Lateral LR'];
            elseif idata == 3
                filename = ['Sub' isubject '_AxLR.exp'];
                fname = ['Subject ' isubject ' Axial LR'];
            elseif idata == 4
                filename = ['Sub' isubject '_Task01.exp'];
                fname = ['Subject ' isubject '  Standing Crate
wo Rotation'];
            elseif idata == 5
                filename = ['Sub' isubject '_Task02.exp'];
                fname = ['Subject ' isubject ' Standing Crate 45
L Rotation'];
            elseif idata == 6
                filename = ['Sub' isubject '_Task03.exp'];
                fname = ['Subject ' isubject ' Standing Crate 45
R Rotation'];
            elseif idata == 7
                filename = ['Sub' isubject '_Task04.exp'];
                fname = ['Subject ' isubject ' Seated Crate wo
Rotation'];
            elseif idata == 8
                filename = ['Sub' isubject '_Task05.exp'];
                fname = ['Subject ' isubject ' Seated Crate 45 L
Rotation'];
```

```
            else
                filename = ['Sub' isubject '_Task06.exp'];
                fname = ['Subject ' isubject ' Seated Crate 45 R
Rotation'];
            end


            %% Define Rotation Sequence

            if seqnum == 1
                rotseq = 'XYZ';
            elseif seqnum == 2
                rotseq = 'XZY';
            elseif seqnum == 3
                rotseq = 'YXZ';
            elseif seqnum == 4
                rotseq = 'YZX';
            elseif seqnum == 5
                rotseq = 'ZXY';
            else
                rotseq = 'ZYX';
            end

            rotationsequence = cat(2, rotseq(1), rotseq(2),
rotseq(3));

            %% Data allocation

            data = dlmread([path filename],'',10,0);

            Man = data(:,2:4);   % X, Y and Z position of Sensor
2 located on the menubrium, tail down
            ManR = data(:,5:13);  % Rotation matrices for sensor
2 located at the menubrium, tail down
            Thor = data(:,14:16);  % X, Y and Z position of
Sensor 3 located on the lumbar, tail down
            ThorR = data(:,17:25); % Rotation matrices for
sensor 3 located on the lumbar, tail down
            Sac = data(:,26:28);   % X, Y and Z position of
sensor 4 located on the sacrum, tail down
            SacR = data(:,29:37);  % Rotation matrices for
sensor 4 located on the sacrum, tail down


            %% Animation of sensors: Check
%             plotaxes(fname, Sac, SacR, Thor, ThorR, Man,
ManR); % plotaxes is a called function
```

```
            %% Correct for out of range sensors (position)

            i = Man.^2 ;
            ii = Thor.^2;
            iii = Sac.^2;

            j = find(sum(i,2) >= 1);
            Man(j,:) = NaN;

            jj = find(sum(ii,2) >= 1);
            Thor(jj,:) = NaN;

            jjj = find(sum(iii,2) >= 1);
            Sac(jjj,:) = NaN;

            for k = 1:size(i,1)
                if Man(k,1) == 0 && Man(k,2) == 0 && Man(k,3) ==
0
                    Man(k,:) = NaN;
                end
            end

            for kk = 1:size(ii,1)
                if Thor(kk,1) == 0 && Thor(kk,2) == 0 &&
Thor(kk,3) == 0
                    Thor(kk,:) = NaN;
                end
            end

            for kkk = 1:size(iii,1)
                if Sac(kkk,1) == 0 && Sac(kkk,2) == 0 &&
Sac(kkk,3) == 0
                    Sac(kkk,:) = NaN;
                end
            end


            %% Rotation Corrections
            initmat = [0 0 -1; 0 -1 0; -1 0 0]; % Orientation
correction matrix

            %%% Apply orientation correction
            ManRcorrinit = rotatecorr(ManR,inv(initmat));
            SacRcorrinit = rotatecorr(SacR,inv(initmat));
            ThorRcorrinit = rotatecorr(ThorR,inv(initmat));

            %%% Re-orient to subject orientation (via pelvic
torsion)
```

```
            pelvic_torsion = 0;

            [ManRcorr] = rotateback_once(ManRcorrinit,
pelvic_torsion);
            [SacRcorr] = rotateback_once(SacRcorrinit,
pelvic_torsion);
            [ThorRcorr] = rotateback_once(ThorRcorrinit,
pelvic_torsion);


             %% Animation of sensors after orientation fix:
Check
%             plotaxes(fname, Sac, SacRcorr, Thor, ThorRcorr,
Man, ManRcorr);


            %% Convert rotation matrix data into euler angles
            % ManE = manubrium Euler angles
            % ThorE = thoracic (T10) Euler angles
            % SacE = sacrum (S1) Euler angles

            [ManE] = MtoE2_dcm2angle(ManRcorr,rotationsequence);
            [ThorE] =
MtoE2_dcm2angle(ThorRcorr,rotationsequence);
            [SacE] = MtoE2_dcm2angle(SacRcorr,rotationsequence);

            % Convert radians to degrees
            ManE = ManE./pi.*180;
            ThorE = ThorE./pi.*180;
            SacE = SacE./pi.*180;

            %% Build virtual torso sensor from T10, S1, and
manubrium sensors
            [torsoM] = torsoMatrix6(Thor, Sac, Man);

            % Orientation corrections on virtual torso sensor
            torsoinit = [-1 0 0; 0 1 0; 0 0 1];
            [torsoMcorrinit] =
rotatecorr(torsoM,inv(torsoinit));

            %%% If re-orienting with constant pelvic torsion
            [torsoMcorr] = rotateback_once(torsoMcorrinit,
pelvic_torsion);

            % Calculate Euler angles for torso sensor
            [torsoE] =
MtoE2_dcm2angle(torsoMcorr,rotationsequence);
```

```
            torsoE = torsoE./pi.*180;

            %% Calculate lumbar angles, torso flexion angles,
torso lateral bending, pelvic tilt, lumbar torsion, and lumbar
lateral bending

            [flexion, lumbar, torsion, lateral, torlat] =
LumFlexTor_06(SacE, ThorE, torsoE,
rotationsequence);

            %% Create matrix for statistics use (means, standard
deviations, analysis in R)

            [piflex, pilum, pilat, pitor, pflex, plum, plat,
ptor, pcycle, stdflex, stdlum, stdlat, stdtor] =
StatCycle(flexion, lumbar, lateral, torsion);

            %%% Single (representative) subject plot
            c = ['r' 'g' 'b' 'c' 'm' 'k'];

            xind = strfind(rotationsequence,'X');
            yind = strfind(rotationsequence,'Y');
            zind = strfind(rotationsequence,'Z');


%             %%%% 2x3 T10 and S1 sensor plot (to be used for
code running with one subject)
%             figure(idata*10000);
%             subplot(2,3,1);
%             plot(ThorE(:,xind),c(seqnum));
%             hold on
%             title(['T10 X rotation: ' fname]);
%
%             subplot(2,3,2);
%             plot(ThorE(:,yind),c(seqnum));
%             hold on
%             title(['T10 Y rotation: ' fname]);
%
%             subplot(2,3,3);
%             plot(ThorE(:,zind),c(seqnum));
%             hold on
%             title(['T10 Z rotation: ' fname]);
%
%             subplot(2,3,4);
%             plot(SacE(:,xind),c(seqnum));
%             hold on
%             title(['S1 X rotation: ' fname]);
%
```

```
%             subplot(2,3,5);
%             plot(SacE(:,yind),c(seqnum));
%             hold on
%             title(['S1 Y rotation: ' fname]);
%
%             subplot(2,3,6);
%             plot(SacE(:,zind),c(seqnum));
%             hold on
%             title(['S1 Z rotation: ' fname]);
%
%
%             set(gcf, 'Position', get(0, 'Screensize'));


            if isub == 2 && idata == 4 && seqnum == 1
                %                 start = 1;
                %                 nextstart = 10;

                start2 = 1;
                nextstart2 = 100;
            else
                %                 start = nextstart + 1;
                %                 nextstart = start + 9;

                start2 = nextstart2 + 1;
                nextstart2 = start2 + 99;
            end

            %% Matrix for comparing Euler angles during one lift
cycle across sequences

            rawcycle(start2:nextstart2,1) = isub;
            rawcycle(start2:nextstart2,2) = seqnum;
            rawcycle(start2:nextstart2,3) = idata;
            rawcycle(start2:nextstart2,4) = 1:100;
            rawcycle(start2:nextstart2,5) = piflex;
            rawcycle(start2:nextstart2,6) = pilum;
            rawcycle(start2:nextstart2,7) = pilat;
            rawcycle(start2:nextstart2,8) = pitor;
            rawcycle(start2:nextstart2,9) = status;
            rawcycle(start2:nextstart2,10) = gen;

        end

    end

end
```

```
xlswrite('08292018_data.xlsx',rawcycle);


%% Save statistics matrix (for interaction plots)

% xlswrite('statsmat2.xlsx',statsmat2);

%% Averaging percent value across subjects for each sequence and
task

% % statsmat = csvread('statsmat2.csv'); %%% read in csv file
into matlab matrix

rawcycle(isnan(rawcycle)) = 0;

for dd = 4:9 %task
    for ss = 1:6 % sequence

        %%% Set up count for growing rows; one cycle is divided
into ten
        %%% 10% increments, so each iteration of the for loop
must step by
        %%% 10s

        for ii = 1:size(rawcycle,1)
            if rawcycle(ii,3) == dd && rawcycle(ii,2) == ss %
pull from rawcycle matrix all the rows for a specific
task+sequence combination
                subcount = rawcycle(ii,1); % sub count =
tabulating by subject number
                tempflex(rawcycle(ii,4),subcount) =
rawcycle(ii,5);
                templum(rawcycle(ii,4),subcount) =
rawcycle(ii,6);
                templat(rawcycle(ii,4),subcount) =
rawcycle(ii,7);
                temptor(rawcycle(ii,4),subcount) =
rawcycle(ii,8);
            end
        end

        %%% For instances where the values = 0, change to NaN
        tempflex(tempflex == 0) = NaN;
        templum(templum == 0) = NaN;
        templat(templat == 0) = NaN;
        temptor(temptor == 0) = NaN;

        for pp = 1:100 % For each step of the lift cycle
```

```
            tempmean(pp,1) = dd;
            tempmean(pp,2) = ss;
            tempmean(pp,3) = pp;

            %%% Calculate means and standard deviations,
ignoring NaNs

            tempmean(pp,4) = nanmean(tempflex(pp,:),2);
            tempmean(pp,5) = nanstd(tempflex(pp,:),1,2);

            tempmean(pp,6) = nanmean(templum(pp,:),2);
            tempmean(pp,7) = nanstd(templum(pp,:),1,2);

            tempmean(pp,8) = nanmean(templat(pp,:),2);
            tempmean(pp,9) = nanstd(templat(pp,:),1,2);

            tempmean(pp,10) = nanmean(temptor(pp,:),2);
            tempmean(pp,11) = nanstd(temptor(pp,:),1,2);

        end

        rawmeans(:,:,dd-3,ss) = tempmean;
    end
end
%
% % Save variables for means and full cycle (for analysis in R)
%
% % save RotLiftWorkspace rawmeans rawcycle
%
% %% Plots based on rawmeans

for j = 1:6 % task

    c = ['r' 'g' 'b' 'c' 'm' 'k'];

    if j == 1
        fname = 'Standing Crate wo Rotation';
    elseif j == 2
        fname = 'Standing Crate 45 L Rotation';
    elseif j == 3
        fname = 'Standing Crate 45 R Rotation';
    elseif j == 4
        fname = 'Seated Crate wo Rotation';
    elseif j == 5
        fname = 'Seated Crate 45 L Rotation';
    else
        fname = 'Seated Crate 45 R Rotation';
```

```
    end

        for k = [4,6,8,10] % Euler angle
            if k == 4
                angle = 'Extension (-) / Flexion (+)';
            elseif k == 6
                angle = 'More Lordotic (-) / More Kyphotic (+)';
            elseif k == 8
                angle = 'Left (-) / Right (+)';
            else
                angle = 'Right (-) / Left (+)';
            end

    for m = 1:6 % sequence

            %%% Plotting means

                    fnum = 1000*j + k;
                    fnum = 1000*j;

                    figure(fnum);
                    subplot(1,4,(k/2)-1);
                    plot(rawmeans(:,3,j,m),rawmeans(:,k,j,m),c(m
));
                    xlabel('Percent Lift Cycle');
                    ylabel(angle);
                    hold on

                    %%% Plotting std

%                     fnum2 = 1000*j + 9;
%                     figure(fnum2);
%                     subplot(1,4,(k/2)-1);
%                     plot(rawmeans(:,3,j,m),rawmeans(:,k+1,j,m)
,c(m));
%                     xlabel('Percent Lift Cycle');
%                     ylabel([angle ' Std']);
%                     hold on


                end

%                   legend('XYZ','XZY','YZX','ZYX');

%                   legend('XYZ','XZY','YXZ','YZX','ZXY','ZYX');
%
                axes('Position', [0 0 0.35 0.96]
,'Visible','off');
```

```
                text(0.5, 0.98,'Trunk Flexion', 'FontSize', 12);

                axes('Position', [0 0 0.75 0.96]
,'Visible','off');
                text(0.5, 0.98,'Lumbar Flexion', 'FontSize',
12);

                axes('Position', [0 0 1.20 0.96]
,'Visible','off');
                text(0.5, 0.98,'Lateral Bending', 'FontSize',
12);

                axes('Position', [0 0 1.63 0.96]
,'Visible','off');
                text(0.5, 0.98,'Torsion', 'FontSize', 12);

                axes('Position',[0 0 1 1],'Visible','off');
                text(0.35, 0.98,['Euler Angles For ' fname],
'FontSize', 15,'FontWeight','bold');
                hold on

            end

%         figure(j);
%         plot(rawmeans(:,4,j,m),rawmeans(:,6,j,m));
%         hold on
%         legend('XYZ','XZY','YXZ','YZX','ZXY','ZYX');

        set(gcf, 'Position', get(0, 'Screensize'));

    end
% end



%% Save figures

% figpath = ['/Volumes/LAB
DRIVE/RotLift/Plots_SubRotSeq/FigPlots/AllSub v11Task'
num2str(idata) '/']; % Define directory path for where .fig
files will be saved
tiffpath = ['/Volumes/LAB DRIVE/RotLift/Plots_SubRotSeq/2018-08-
29 Figures/']; % Define directory path for where .tiff files
will be saved
figHandles = get(0,'Children'); % Finds all figure handles
%             set(figHandles,'Visible','on'); % Makes previously
suppressed figures visible again; WARNING: This will open
EVERYTHING
```

```
for i=1:length(figHandles)
    fig = get(figHandles(i),'Number');
%     saveas(figHandles(i), fullfile(figpath,[num2str(fig)
'.fig'])); % Saves figure as a .fig file
    saveas(figHandles(i), fullfile(tiffpath,[num2str(fig)
'.tiff'])); % Saves figure as a .tiff file
end

close all

%% For opening specific set of figures:
% for figset = [1111] % define your set of figures
%     set(figset,'Visible','on');
% end
```

Appendix A.2: Raw data plot check

```
% Author: Sara Wilson
% Revision date:
% Program: plotaxes.m
% Purpose: Plots and animates raw sensor data to check initial
axes orientations

function [] = plotaxes(fname, Sac, SacR, Thor, ThorR, Man, ManR)

figure();
[m,n] = size(Man);
h1=plot3(Man(1,1), Man(1,2), -Man(1,3),'or'); % Plot Manubrium
sensor
grid
hold on
axis([-0.5 1 -1 0.5 -0.5 1]);
h2=plot3(Thor(1,1), Thor(1,2), -Thor(1,3),'og'); % Plot T10
sensor
h3=plot3(Sac(1,1), Sac(1,2), -Sac(1,3),'ob'); % Plot S1 sensor

h4=plot3([Man(1,1) Man(1,1)+ManR(1,1)/10], [Man(1,2)
Man(1,2)+ManR(1,2)/10], -[Man(1,3) Man(1,3)+ManR(1,3)/10],'-r');
h5=plot3([Man(1,1) Man(1,1)+ManR(1,4)/10], [Man(1,2)
Man(1,2)+ManR(1,5)/10], -[Man(1,3) Man(1,3)+ManR(1,6)/10],'-g');
h6=plot3([Man(1,1) Man(1,1)+ManR(1,7)/10], [Man(1,2)
Man(1,2)+ManR(1,8)/10], -[Man(1,3) Man(1,3)+ManR(1,9)/10],'-b');

h7=plot3([Thor(1,1) Thor(1,1)+ThorR(1,1)/10], [Thor(1,2)
Thor(1,2)+ThorR(1,2)/10], -[Thor(1,3)
Thor(1,3)+ThorR(1,3)/10],'-r');
h8=plot3([Thor(1,1) Thor(1,1)+ThorR(1,4)/10], [Thor(1,2)
Thor(1,2)+ThorR(1,5)/10], -[Thor(1,3)
Thor(1,3)+ThorR(1,6)/10],'-g');
h9=plot3([Thor(1,1) Thor(1,1)+ThorR(1,7)/10], [Thor(1,2)
Thor(1,2)+ThorR(1,8)/10], -[Thor(1,3)
Thor(1,3)+ThorR(1,9)/10],'-b');

h10=plot3([Sac(1,1) Sac(1,1)+SacR(1,1)/10], [Sac(1,2)
Sac(1,2)+SacR(1,2)/10], -[Sac(1,3) Sac(1,3)+SacR(1,3)/10],'-r');
```

```
h11=plot3([Sac(1,1) Sac(1,1)+SacR(1,4)/10], [Sac(1,2)
Sac(1,2)+SacR(1,5)/10], -[Sac(1,3) Sac(1,3)+SacR(1,6)/10],'-g');
h12=plot3([Sac(1,1) Sac(1,1)+SacR(1,7)/10], [Sac(1,2)
Sac(1,2)+SacR(1,8)/10], -[Sac(1,3) Sac(1,3)+SacR(1,9)/10],'-b');

hold off
% legend('Manubrium','Thoracic','Sacrum');
title ([fname,' Sensor Positions']);
xlabel ('X position');
ylabel ('Y position');
zlabel ('Height');

for i = 1:m
    set(h1,'XData',Man(i,1));
    set(h1,'YData',Man(i,2));
    set(h1,'ZData',-Man(i,3));
    set(h2,'XData',Thor(i,1));
    set(h2,'YData',Thor(i,2));
    set(h2,'ZData',-Thor(i,3));
    set(h3,'XData',Sac(i,1));
    set(h3,'YData',Sac(i,2));
    set(h3,'ZData',-Sac(i,3));
    set(h4,'XData',[Man(i,1) Man(i,1)+ManR(i,1)/10]);
    set(h4,'YData',[Man(i,2) Man(i,2)+ManR(i,2)/10]);
    set(h4,'ZData',-[Man(i,3) Man(i,3)+ManR(i,3)/10]);
    set(h5,'XData',[Man(i,1) Man(i,1)+ManR(i,4)/10]);
    set(h5,'YData',[Man(i,2) Man(i,2)+ManR(i,5)/10]);
    set(h5,'ZData',-[Man(i,3) Man(i,3)+ManR(i,6)/10]);
    set(h6,'XData',[Man(i,1) Man(i,1)+ManR(i,7)/10]);
    set(h6,'YData',[Man(i,2) Man(i,2)+ManR(i,8)/10]);
    set(h6,'ZData',-[Man(i,3) Man(i,3)+ManR(i,9)/10]);

    set(h7,'XData',[Thor(i,1) Thor(i,1)+ThorR(i,1)/10]);
    set(h7,'YData',[Thor(i,2) Thor(i,2)+ThorR(i,2)/10]);
    set(h7,'ZData',-[Thor(i,3) Thor(i,3)+ThorR(i,3)/10]);
    set(h8,'XData',[Thor(i,1) Thor(i,1)+ThorR(i,4)/10]);
    set(h8,'YData',[Thor(i,2) Thor(i,2)+ThorR(i,5)/10]);
    set(h8,'ZData',-[Thor(i,3) Thor(i,3)+ThorR(i,6)/10]);
    set(h9,'XData',[Thor(i,1) Thor(i,1)+ThorR(i,7)/10]);
    set(h9,'YData',[Thor(i,2) Thor(i,2)+ThorR(i,8)/10]);
    set(h9,'ZData',-[Thor(i,3) Thor(i,3)+ThorR(i,9)/10]);
```

```
    set(h10,'XData',[Sac(i,1) Sac(i,1)+SacR(i,1)/10]);
    set(h10,'YData',[Sac(i,2) Sac(i,2)+SacR(i,2)/10]);
    set(h10,'ZData',-[Sac(i,3) Sac(i,3)+SacR(i,3)/10]);
    set(h11,'XData',[Sac(i,1) Sac(i,1)+SacR(i,4)/10]);
    set(h11,'YData',[Sac(i,2) Sac(i,2)+SacR(i,5)/10]);
    set(h11,'ZData',-[Sac(i,3) Sac(i,3)+SacR(i,6)/10]);
    set(h12,'XData',[Sac(i,1) Sac(i,1)+SacR(i,7)/10]);
    set(h12,'YData',[Sac(i,2) Sac(i,2)+SacR(i,8)/10]);
    set(h12,'ZData',-[Sac(i,3) Sac(i,3)+SacR(i,9)/10]);
    pause(0.005);

%     frame = getframe(gcf);
%     writeVideo(writerObj,frame);
end
```

Appendix A.3: Orientation corrections

rotatecorr.m

```
% Author: Sara Wilson
% Program: rotatecorr.m
% Purpose: This function adjusts the orientation of a sensor's
raw Motion Monitor quaternion rotation matrix by a given
correction matrix

function [matrixnew] = rotatecorr(matrix,correction)

for i = 1:length(matrix)

   A = [matrix(i,1) matrix(i,2) matrix(i,3); matrix(i,4)
matrix(i,5) matrix(i,6); matrix(i,7) matrix(i,8) matrix(i,9)];
    A = A*correction;
    matrixnew(i,:) = [A(1,1) A(1,2) A(1,3) A(2,1) A(2,2) A(2,3)
A(3,1) A(3,2) A(3,3)];

end
end
```

rotateback_cont.m

```
% Author: Sara Wilson
% Program: rotateback_once.m
% Purpose: This function adjusts a sensor's rotation matrix to
be oriented relative to a constant value representing the sacral
marker's z-axis rotation (direction facing)

function [matrixnew] = rotateback_cont(matrix, pelvic_torsion)

pelvic_torsion = pelvic_torsion/180*pi;

for i = 1:length(matrix)
    rotateback = inv([cos(pelvic_torsion) -sin(pelvic_torsion)
0; sin(pelvic_torsion) cos(pelvic_torsion) 0; 0 0 1]);
    A = [matrix(i,1) matrix(i,2) matrix(i,3); matrix(i,4)
matrix(i,5) matrix(i,6); matrix(i,7) matrix(i,8) matrix(i,9)];
    A = rotateback*A;
```

```
    matrixnew(i,:) = [A(1,1) A(1,2) A(1,3) A(2,1) A(2,2) A(2,3)
A(3,1) A(3,2) A(3,3)];

end
```

Appendix A.4: Rotation matrix conversion to Euler angles

```
% Authors: Timothy Craig, Sara Wilson
% Revision Date 9/20/2016
% For use in convering Motion Monitor data collected in matrix
form


function [dataout] = MtoE2_dcm2angle(matrix, rotationsequence)
% Converts instant 3D position matrix to EULER
ANGLES
%   This function works well when data has been exported from
Motion
%   Monitor in matrix form [M00, M01 M02; M10 M11 M12; M20 M21
M22]
    % No need to separate matrices!

for i = 1:length(matrix)
    A(1,1) = matrix(i,1);
    A(1,2) = matrix(i,2);
    A(1,3) = matrix(i,3);
    A(2,1) = matrix(i,4);
    A(2,2) = matrix(i,5);
    A(2,3) = matrix(i,6);
    A(3,1) = matrix(i,7);
    A(3,2) = matrix(i,8);
    A(3,3) = matrix(i,9);
    [angleseuler(i,1) angleseuler(i,2) angleseuler(i,3)] =
dcm2angle(A,rotationsequence);
end

dataout = angleseuler;
end
```

Appendix A.5: Torso sensor construction

torsoMatrix6.m

```
% Edited by: Ednah Louie
% Revision Date: 02/28/2018
% Program: torsoMatrix6.m
% Purpose: Converts raw Motion Monitor instant 3D position data
to create a virtual torso sensor representing the torso
orientation using vector differences and cross product
multiplication. This function replicates the calculations from
the Motion Monitor program where feedback is provided for the
lifting task.

function [torsoM] = torsoMatrix6(Thor, Sac, Man)

warning off

for i=1:length(Sac)

    a(i,:) = (Thor(i,:) - Sac(i,:))./norm(Thor(i,:) - Sac(i,:));
    b(i,:) = (Man(i,:) - Thor(i,:))./norm(Man(i,:)-Thor(i,:));
    c(i,:) = cross(a(i,:),b(i,:));
    cc(i,:) = c(i,:)./norm(c(i,:));

    A(1,:) = cc(i,:);
    A(2,:) = -cross(c(i,:),a(i,:))./norm(cross(c(i,:),a(i,:)));
    A(3,:) = -a(i,:);

    A = inv(A);

    M(i,1:3) = A(1,:);
    M(i,4:6) = A(2,:);
    M(i,7:9) = A(3,:);

end

torsoM = M; % This output can now be converted to Euler Angles

end
```

Appendix A.6: Calculation of Euler angles

```
% Edited by: Ednah Louie
% Revision date: 05/30/2017
% Program: LumFlexTor_06.m
% Purpose: Calculates spine mechanics based on the Euler angles
of each sensor.

function [flexion, lumbar, torsion, lateral, torlat] =
LumFlexTor_06(SacEfix, ThorEfix, torsoEfix, rotationsequence)

% Identify the position of the rotation axis for each Euler
angle in the rotationsequence string. Ex: For a rotationsequence
XYZ, the X index = 1, Y = 2, Z = 3.

flexind = strfind(rotationsequence,'X');
latind = strfind(rotationsequence,'Y');
lumind = strfind(rotationsequence,'X');
torind = strfind(rotationsequence,'Z');

% Euler angles are calculated
flexion = torsoEfix(:,flexind); %torso flexion
lumbar = (ThorEfix(:,lumind) - SacEfix(:,lumind)); %lumbar
(measure of kyphotic and lordotic; the curvature of the lower
back)
torsion = ThorEfix(:,torind) - SacEfix(:,torind); %torsion
(measure of lumbar axial twist)
torlat = torsoEfix(:,latind); %lateral (measure of torso lateral
bending)
lateral = ThorEfix(:,latind) - SacEfix(:,latind);

% Applies a fix for gimbal locks of 90 and/or 180 degree at the
very start of data collection
torsion = jump90(torsion);
torlat = jump90(torlat);
lateral = jump90(lateral);
lumbar = jump90(lumbar);

end
```

Appendix A.7: Data selection for one lift cycle

```
% Author: Ednah Louie
% Revision date: 04/18/2018
% Program: StatCycle.m
% Purpose: Takes calculated spine angles and, for each one:
% -- truncates it into the time duration of one lift cycle
% -- resizes one lift cycle to 100 data point
% -- calculates averages for every 10% of a lift cycle

function [piflex, pilum, pilat, pitor, pflex, plum, plat, ptor,
pcycle, stdflex, stdlum, stdlat, stdtor] = StatCycle(flexion,
lumbar, lateral, torsion)

%%% Divide flexion into thirds, then take the first clean cycle
in the second third

[m,~] = size(flexion);
third = round(m/3); %% divided flexion into three time groups
divflex = flexion(third:third*2,:); %% selects second of three
time groups (so the single cycle is taken from the middle of the
whole task trial)

%%% Test if there is a string of NaNs in divflex, which will
throw off findpeaks

[a,~] = find(isnan(divflex));

if isempty(a) == 1

    [~, locs] =
findpeaks(divflex,'MinPeakHeight',5,'MinPeakDistance',500); %
find peaks, minimum peak distance 500 time points and minimum
peak height of 5 degrees

    pkloc = [third+locs(1),third+locs(2)]; % takes the first
full cycle based on peaks

    %%% Truncate spine angles to represent one lift cycle
    trunflex = flexion(pkloc(1):pkloc(2),:); %truncated flexion
    trunlum = lumbar(pkloc(1):pkloc(2),:); %truncated lumbar
```

```
    trunlat = lateral(pkloc(1):pkloc(2),:); %truncated lateral
    truntor = torsion(pkloc(1):pkloc(2),:); %truncated torsion

    %%% interpolate so one lift cycle = 100 data points
    lengthcycle = length(trunflex);
    x_new = 1:1:100;
    fdata = 100/lengthcycle:100/lengthcycle:100;
    piflex = interp1(fdata,trunflex,x_new)';
    pilum = interp1(fdata,trunlum,x_new)';
    pilat = interp1(fdata,trunlat,x_new)';
    pitor = interp1(fdata,truntor,x_new)';

else
    piflex(1:100,1) = NaN;
    pilum(1:100,1) = NaN;
    pilat(1:100,1) = NaN;
    pitor(1:100,1) = NaN;
end

%%% Take the average across each set of 10 data points to get an
average spine angle for each percent of the lift cycle

tab = 10:10:100; % represents every 10% of lift cycle
for i = 1:10
    if i == 1
        pflex(i,1) = nanmean(piflex(1:tab(i),1));
        plum(i,1) = nanmean(pilum(1:tab(i),1));
        plat(i,1) = nanmean(pilat(1:tab(i),1));
        ptor(i,1) = nanmean(pitor(1:tab(i),1));

        stdflex(i,1) = nanstd(piflex(1:tab(i),1));
        stdlum(i,1) = nanstd(pilum(1:tab(i),1));
        stdlat(i,1) = nanstd(pilat(1:tab(i),1));
        stdtor(i,1) = nanstd(pitor(1:tab(i),1));
    else
        pflex(i,1) = nanmean(piflex(tab(i-1):tab(i),1));
        plum(i,1) = nanmean(pilum(tab(i-1):tab(i),1));
        plat(i,1) = nanmean(pilat(tab(i-1):tab(i),1));
        ptor(i,1) = nanmean(pitor(tab(i-1):tab(i),1));

        stdflex(i,1) = nanstd(piflex(tab(i-1):tab(i),1));
        stdlum(i,1) = nanstd(pilum(tab(i-1):tab(i),1));
```

```
        stdlat(i,1) = nanstd(pilat(tab(i-1):tab(i),1));
        stdtor(i,1) = nanstd(pitor(tab(i-1):tab(i),1));
    end
end


pflex;
plum;
plat;
ptor;
pcycle = tab';
stdflex;
stdlum;
stdlat;
stdtor;

piflex;
pilum;
pilat;
pitor;
```